DETECTION PROCESSES BASED ON THE VITERBI ALGORITHM

by

J P DRISCOLL, B.Sc., M.Sc.

A Doctoral Thesis
Submitted in partial fulfilment of the
requirements for the award of
Doctor of Philosophy at
Loughborough University of Technology

August 1979

Supervisors: A P Clark
Department of Electronic & Electrical Engineering
and
G W Irwin
Department of Engineering Mathematics

ABSTRACT

A number of detection processes are proposed, which are developments of the Viterbi Algorithm (V.A) detector. These detectors are suitable for use in high speed digital data transmission systems, in which the associated baseband channel introduces severe intersymbol interference.

The aim of the project has been to develop algorithms for detection processes, which do not require the large amount of computation that is sometimes needed by the V.A. These algorithms should ideally have performances which are close to optimum, and should not be significantly more complicated to implement than the V.A.

The proposed detectors are compared to the V.A. detector and a conventional non linear equalizer, by means of computer simulation tests. The tolerance to additive white Gaussian noise is given for each detector, when used with a number of time invariant channels. Graphs are given showing the variation in the performances of the detectors, with certain system parameters.

Consideration is given to channels whose response grows slowly with time, so that the first few components of their sampled impulse responses are small. Results are presented which give an indication of how small these initial elements have to be, for them to be profitably ignored by the detector.

It was found that some of the detectors had a tendency to become fixed in a "poor" mode of operation, if they were left running for a sufficiently long time. In this mode, some of the vectors stored by the system are identical, and the effective number of these vectors is reduced, thus causing a loss of performance. The occurrence of this "poor" mode of operation is analysed, and some modifications to the detectors are suggested, to overcome the problem.

ACKNOWLEDGEMENTS

The author would like to express his gratitude to his supervisors, A P Clark, G W Irwin and the late J W Worster, for their help and guidance during the project.

The financial support of the Science Research Council, which made the project possible, is gratefully acknowledged.

The author would like to thank those who have shown a continuing interest in the progress of the work, and to thank Janet Smith for typing this thesis.

Many thanks also for the facilities provided by the Department of Engineering Mathematics, Loughborough University.

CONTENTS

Page No

Page No.

CHAPTER 4:

Glossary of Symbols and Terms

V.A.	Viterbi Algorithm.
$\simeq$	Approximately equal to.
$\triangleq$	Defined to be equal to.
$\prod_{i=0}^{n} x_i$	The product $x_0\ x_1\ x_2 \dots x_n$.
$\ell n(x)$ or $\ell n\ x$	The natural logarithm of x.
$\lvert\underline{x}\rvert$	The modulus of the vector $\underline{x}$, i.e. the Euclidean distance between $\underline{x}$ and the origin.
$N(\mu,\sigma^2)$	The normal probability distribution with mean μ and variance σ^2.
$[\underline{V}]_g$	The vector formed from the last g components of $\underline{V}$, for any given vector $\underline{V}$.
$(\underline{V}_i,J)$	The vector whose last component is J and whose other components are those of the vector $\underline{V}_i$.
m	The number of possible values of a data element.
g+1	The number of components of a channel's sampled impulse response.
σ^2	The variance of a normally distributed random variable representing noise.
T	The interval of time between the transmission of successive data elements.
s_i	A data element.
$\{x_i\}$	The sequence of numbers x_0, x_1, x_2 ...
y(t)	The impulse response of a transmission channel.
y_i	A sample value of the impulse response y(t).
$\underline{Y}$	The vector representing a channel's sampled impulse response.
w(t)	A function representing white Gaussian noise.
w_i	A sample of the function w(t) representing white Gaussian noise.
r(t)	The signal received at the detector.

r_i A sample of the received signal $r(t)$.

$\delta(t)$ A unit impulse at time 0.

(I,J) The vector with components I and J.

$(I,J)_k$ The node representing the possible vector value (I,J) of $\underline{Q}_k$, on a trellis diagram.

$\{\underline{P}_i(k,I,J)\}$ The sequence of k+1 vectors giving the shortest path through the trellis, from point A to the node $(I,J)_k$. Each vector in this sequence is a function of k, I and J.

$\{\underline{P}_i(k-1,K,I)\},(I,J)$ The sequence of k+1 vectors whose first k members are those of the sequence $\{\underline{P}_i(k-1,K,I)\}$, and whose k+1 st. member is the vector (I,J).

$u_k(I,J)$ The length of the shortest path through the trellis, from point A to the node $(I,J)_k$.

$d_k^{(2)}[(K,I),(I,J)]$ The distance between the two trellis nodes $(K,I)_k$ and $(I,J)_{k+1}$.

k The number of N component vectors stored at the start of each cycle.

N The number of components of the vectors stored at the start of each cycle.

$\{\underline{Q}_i\}$ The sequence of vectors $\underline{Q}_0$, $\underline{Q}_1$, $\underline{Q}_2$, ...

$\underline{Q}(I)$ The I th. N component vector stored at the start of some unspecified cycle of a detection process.

$[\underline{Q}(I),J]$ The N+1 component vector whose first N components are those of $\underline{Q}(I)$ and whose N+1 st. component is J.

$\underline{I}(I,J)$ Same as $[\underline{Q}(I),J]$.

$\underline{R}(K)$ One of the N+1 component vectors of the form $\underline{I}(I,J)$, which have been selected by the decision rule for the process.

$u(I)$ The cost for the vector $\underline{Q}(I)$.

$V(I,J)$ The cost for the vector $\underline{I}(I,J)$.

$[\underline{Q}(I),J]_{g+1}$ The vector formed from the last g+1 components of vector $[\underline{Q}(I),J]$.

$[\underline{T}(I,J)]_{g+1}$ Same as $[\underline{Q}(I),J]_{g+1}$.

$\underline{Y}.[\underline{T}(I,J)]_{g+1}$ The scalar product of the vectors $\underline{Y}$ and $[\underline{T}(I,J)]_{g+1}$.

$[\underline{Q}(I),J,L]$ The N+2 component vector whose first N components are those of $\underline{Q}(I)$ and whose last two components are J and L.

$D(I,J,L)$ The cost for the vector $[\underline{Q}(I),J,L]$.

$\underline{Q}_j(I)$ The I th. N component vector stored at the start of the j+2 nd. cycle of a detection process.

$[\underline{Q}_j(I),x_{j+1}]$ The N+1 component vector whose first N components are those of $\underline{Q}_j(I)$, and whose N+1 st. component is x_{j+1}.

$\underline{T}_{j+1}(I,x_{j+1})$ Same as $[\underline{Q}_j(I),\ x_{j+1}]$.

$[\underline{T}_{j+1}(I,x_{j+1})]_{g+1}$ The vector formed from the last g+1 components of the vector $\underline{T}_{j+1}(I,\ x_{j+1})$.

$u_j(I)$ The cost for the vector $\underline{Q}_j(I)$.

$V_{j+1}(I,x_{j+1})$ The cost for the vector $\underline{T}_{j+1}(I,x_{j+1})$.

Two sequences $\{a_i\}$ and $\{b_i\}$ are said to be equal if $a_i = b_i$ for all values of i.

CHAPTER 1

1.01 Background

In recent years the amount of data being sent from one point to another, by means of digital signals, has been steadily increasing and this trend seems likely to continue. It is therefore desirable to transmit data as quickly as possible, while making use of existing facilities, in order to keep equipment cost to a minimum.

An essential part of most digital data transmission systems is the detector [2, 11, 12]. The detector takes the received signal, which is usually a distorted version of the signal transmitted, and tries to recover the transmitted signal in a fairly efficient manner.

Many methods of detection already exist [1-49] but some, although simple to implement, give a poor tolerance to noise or do not allow high data transmission rates.

From Shannon's famous channel capacity theorem [1,7], it is theoretically possible to send information over a transmission channel, without errors in detection, at the rate

$$c = W \log_2 (1 + s) \text{ bits/sec} \qquad (1.01)$$

where s is the ratio of signal power to average noise power and W is the bandwidth of the channel. This formula assumes an ideal channel with constant signal attenuation over its bandwidth, and that the noise introduced in the channel is additive white Gaussian noise.

The majority of data transmission is over standard telephone networks which ideally have $W = 2700$ Hz and typically have the signal to noise ratio s given by

$$10 \log_{10} s > 25 \text{ db} \qquad \text{or} \qquad s > 10^{2\cdot 5}$$

Hence, with an optimum detection process and the conditions described above, the transmission rate attainable over such telephone networks has a theoretical maximum value of at least

$$2700 \log_2 (1 + 10^{2\cdot 5}) \simeq 22.4 \times 10^3 \text{ bits/sec}$$

At present, conventional data transmission systems operate at rates of up to 4800 bits/sec so there is clearly much scope for improvement.

1.02 Model of a Data Transmission System

Figure 1.01 describes a theoretical model of a synchronous serial data transmission system. Data is transmitted in the form of a sequence $\{s_i\}$ of numbers, which are used to modulate a sequence of unit impulses $\delta(t)$. The impulses are regularly spaced in time with some interval T seconds between them, so that the sequence $s_0, s_1, \ldots, S_n$ can be represented in the form

$$\sum_{i=0}^{n} s_i \, \delta(t - i\,T)$$

where $\delta(t - i\,T)$ is a unit impulse at time $t = i\,T$. The transmitted signal is assumed to have an even number m of levels, so that each

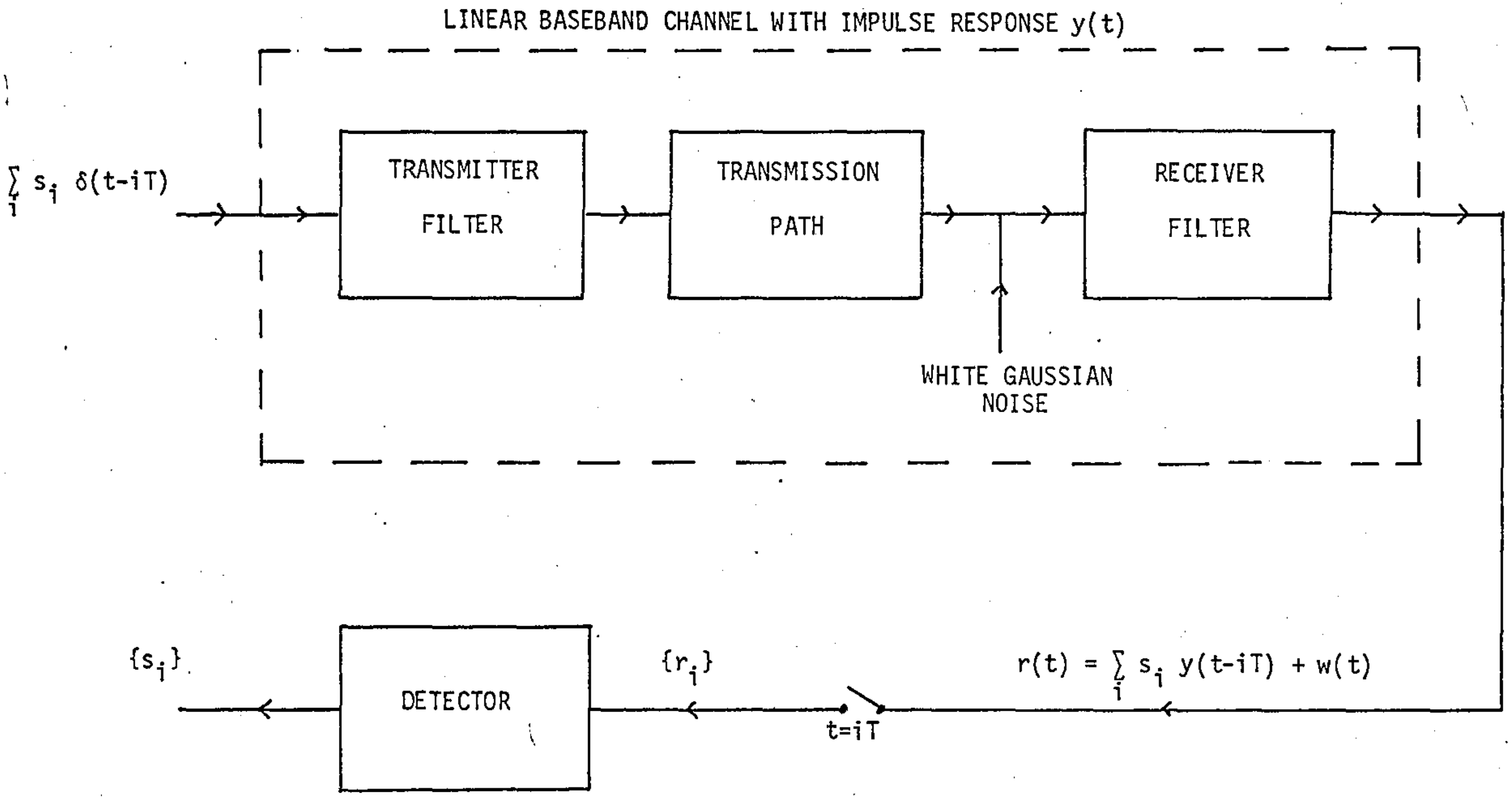

LINEAR BASEBAND CHANNEL WITH IMPULSE RESPONSE y(t)
$\sum_i s_i \, \delta(t-iT)$
TRANSMITTER FILTER
TRANSMISSION PATH
RECEIVER FILTER
WHITE GAUSSIAN NOISE
$\{s_i\}$
DETECTOR
$\{r_i\}$
t=iT
$r(t) = \sum_i s_i \, y(t-iT) + w(t)$

FIGURE 1.01 DATA TRANSMISSION SYSTEM

data element s_i has m possible values. The allowable values for each s_i are ±1, ±3, ±5, ..., ±(m - 1) and each of these values is equally likely. Furthermore, each data element is assumed to be statistically independent of the other data elements.

The transmission path is a linear baseband channel which may include a telephone circuit or a high frequency radio link, in which case a linear modulator at the transmitter and a linear demodulator at the receiver must also be included.

The transmitter filter is used to limit the frequency spectrum of the impulses so that almost all of the energy going into the transmission path is contained in the available bandwidth. It is inefficient to feed the impulses directly to the transmission path, as they have a large (ideally infinite) bandwidth. The transmission path would then cause considerable attenuation of the higher frequency components of the signal, resulting in an unnecessary loss of signal energy.

The model assumes that the only noise introduced by the system, is additive white Gaussian noise, which is introduced between the transmission path and the receiver filter. The other types of additive and multiplicative noise which occur in a practical system are neglected here. It has been shown that the tolerance of a system to additive white Gaussian noise, gives a good guide to its tolerance to other forms of additive noise [6].

The receiver filter cuts out frequencies outside of the bandwidth of the channel, so that much of the noise is eliminated. This filter is assumed to be such that the sample values of the noise function w(t), taken at intervals of T seconds by the sampler, are statistically independent normally distributed random variables, with zero mean and fixed variance. This type of filter may include a noise whitening network. (See reference 6).

The combination of transmitter filter, transmission path and receiver filter, forms a linear baseband channel. The impulse response y(t) of the channel is assumed to be constant, or to vary only slowly with time. If y(t) is not constant, some device for estimating the channel's impulse response must be included at the receiver.

An impulse $\delta(t)$ at the input of the channel causes an output $y(t) + w(t)$, where $w(t)$ is the noise waveform at the output of the receiver filter. The channel is linear so the input

$$\sum_{i=0}^{n} s_i \, \delta(t - i\,T)$$

causes an output given by

$$r(t) = \sum_{i=0}^{n} s_i \, y(t - i\,T) + w(t) \qquad (1.02)$$

r(t) is sampled at intervals of T seconds, and the sequence produced is processed by the detector, to give the sequence $\{s_i^1\}$ which is an estimate of the transmitted sequence $\{s_i\}$. Note that, due to the randomness of w(t), it is not possible to recover the transmitted data sequence with certainty. The best that can be done is to obtain the estimate of $\{s_i\}$ which has the least probability of error or, alternatively, find the estimate whose expected proportion of errors is as small as possible.

1.03 Detection Processes

Usually to attain transmission rates as high as 4800 bits per second, data elements must be transmitted at such a speed, that the response of the system to one of them has not died away before the next one is transmitted. Hence the response of the system at any time may depend on several data elements. This overlapping of signals is known as intersymbol interference. An alternative method of achieving a high rate of data transmission, is to increase the number of signal levels to a point where each data element contains a large amount of information. This avoids the problem of intersymbol interference but usually gives a poorer performance than the former method [9].

A linear equalizer (described in Section 1.10), is among the simplest of detection processes for signals with intersymbol interference, and is an approach which is often used commercially [6, 14-18]. This equalizer can easily be made adaptive to a slowly time varying channel [6, 16, 18], and can be placed either before or after the transmission path. It gives the same tolerance to noise in either of these locations, but a significant improvement can sometimes be obtained by splitting the equalization between the two ends of the path [3, 4].

For channels giving only pure phase distortion (see Section 1.11), the linear equalizer gives the optimum detection process [9, 14]. However, for channels with some amplitude distortion, a non linear detection process can give an improved tolerance to noise.

The nonlinear equalizer, using a feedback transversal equalizer and a process of decision directed cancellation, often offers a better performance than the linear equalizer [6, 14, 23, 24]. When each signal element is detected by the feedback transversal equalizer, its contribution to the next received signal is cancelled, thus removing intersymbol interference. (See Section 1.12). If a data element is incorrectly detected at some stage, the wrong quantity is subtracted in the cancellation process, and errors in the next few elements detected are more likely than they would otherwise be. This effect is known as error extension.

A further improvement in tolerance to noise may be obtained using the system described in Section 1.13, which is a combination of the detector just described and a linear equalizer [6, 14, 24]. Such a combination is sometimes referred to as a non linear equalizer. This arrangement is still not particularly complicated to implement, and can be made to adapt to a slowly time varying channel without great equipment complexity [6].

Where the degree of amplitude distortion is high, more sophisticated detection processes are needed, if a good tolerance to noise is to be obtained. Among these processes is the one described in Section 1.15, in which data is transmitted and received in fairly short sequences, with gaps between them [25-27]. These gaps are large enough so that there is no intersymbol interference between the separate data sequences. Now the optimum detected sequence for each group may be found, by selecting the member from the set of possible data sequences, which minimises some given function. There are no error extension effects with this type of process, as a complete group of data elements is detected at once. The detected sequence for one group, does not then depend on the sequences detected for the previous groups.

A development of the system described above, which makes use of signal cancellation, can be used for the case of continuous data transmission. (See Section 1.16) [29-31]. With this system, a number N of the first received signal samples, are used to produce an estimate of the first N data elements. Only the estimate for the earliest of the elements is taken as a detected element. The contribution of this element is then cancelled from the received signal samples (see Section 1.16), and the process is repeated to detect another data element. The performance of this detection process can approach that of an optimum detector (i.e. one which gives the detected sequence which has the least probability of being in error. [6].

One system which obtains the optimum tolerance to additive white Gaussian noise, is the Viterbi Algorithm (V.A) detector, which is described in Chapter 2. The V.A. was originally proposed, by Viterbi, for decoding convolutional codes. [34]. Sometime later, several authors pointed out that the V.A. could be used as the basis, for a detector in a digital data transmission system with a dispersive channel. [35, 45, 46]. This detector has the disadvantage that, where intersymbol interference exists over a large number of symbols, the amount of computation required can be very large. In this case, the system described in Section 1.16 (employing decision directed cancellation), probably gives a more cost effective process, although its tolerance to noise is usually poorer than that of the V.A. detector.

Several suggestions have been made for modifying the V.A. detector, or using it in conjunction with another detector. It is hoped that, by these means, a system may be found with a reasonable performance and a moderate demand on computation. One proposal was to use a linear filter between the transmission path and the V.A. detector, so that the combined impulse response of the channel and filter is of a fairly short duration. [32, 36]. Unfortunately, the filter usually causes a correlation of the noise samples (see Section 1.10), thus giving some loss of system performance. A suggestion made by Forney was to modify the V.A. so that, instead of an exhaustive search through all possible transmitted sequences, only the sequences which seemed most likely should be considered [33]. A number of systems described in Chapter 3, are based on this idea of Forney's. The systems are also examined in references 47 and 48.

1.04 Outline of the Thesis

Chapter 1, so far, has discussed some of the conventional detection techniques used in digital communication systems. Some of these techniques are discussed in more detail in Sections 1.10 to 1.16.

In the model of a data transmission system described in Section 1.02, the transmission path is a linear baseband channel which may include a telephone circuit or a high frequency radio link. Descriptions of these types of transmission paths, and the types of random noise they suffer from, are therefore given in Sections 1.06 to 1.09.

Chapter 2 gives a detailed description of the V.A., and introduces some of the notation that is used in the later chapters. Maximum likelihood detection is also defined at this point, and it is shown that under certain conditions, the maximum likelihood sequence is the one which has the lowest probability of being in error. It is shown in Section 2.04, that the problem of finding the maximum likelihood detected data sequence, is equivalent to that of finding the shortest path through a given trellis diagram. This shortest path problem is well known, as one that can be solved efficiently with a technique known as dynamic programming. It can then be seen that the V.A. is a dynamic programming algorithm.

In Section 2.09, some indication is given of the number of basic operations required to implement the V.A. It is evident from this section, that the number of such operations can be very large indeed, for cases where the data elements have to be transmitted at high speed, and where the number of signal levels is high.

Chapter 3 begins by considering again, the trellis diagram described in Chapter 2. It is pointed out that some of the possible paths through the trellis, are unlikely to coincide with the optimum path, and may therefore be ruled out without fully assessing their length. Four detection processes (systems 1, 2, 3 and 4) are then described, which are developments of the V.A. detector. These processes were designed, with the hope of cutting down considerably on the amount of computation required by the V.A, while still offering a performance close to that of a maximum likelihood detector.

In Section 3.09, the amount of computation needed per detected data element, is assessed for Systems 1-4. By means of an example, a comparison is then made between the number of basic operations, needed for each of these systems and the V.A. detector.

Section 3.10 gives a number of theorems concerning the operation of Systems 1, 2 and 4, when they are used with a channel whose sampled impulse response has its first component equal to zero. (See Section 2.03 for a definition of the sampled impulse response of a channel).

Chapter 4 deals with the testing of the various detection processes, by means of computer simulation. The reasons for testing the processes in this way are discussed, together with some of the advantages and disadvantages of this method. The simulation results are all from situations, where the detectors are working in the presence of random noise, so the results of the tests are all subject to statistical fluctuation (i.e. if a test is repeated with a difference sequence of noise samples, the result may be changed slightly). Some measure of the confidence in these results is therefore fairly essential. Section 4.04 of the thesis gives a definition of the term, *"Confidence limits"*, and gives an estimate of these limits for some of the simulation tests which follow.

Section 4.06 lists the sampled impulse responses of the various transmission channels, which were used in the tests. A function, 'd', of the impulse response is given, which is known to give a measure of the degree of amplitude distortion introduced by the channels. The modulus and argument of the sampled Fourier Transform is also given, for these sampled impulse responses, so that information about both the amplitude and phase distortions for these channels, can be derived.

Section 4.07 gives the results of simulation tests which compare the performances of each of the Systems 1-4, the V.A. detector and the conventional non linear equalizer. These results cover a fairly wide range of situations, with both a two and a four level signal being used, with each of the channels included in the tests. These simulation tests give a comparison of the various detectors, in terms of the noise power required with them, to give an error rate of 0.004 for a given situation. It would be unreasonable though, to assume that the detection processes with the best performances at this error rate, would be superior over a wide range of error rates. Hence simulation tests were carried out to assess the performance of the systems over one channel, with the proportion of errors occurring, varying from 10^{-1} to 10^{-4}. These tests are described in Section 4.08.

With Systems 1-4, the two main parameters which affect performance and complexity, are the number of vectors stored and the number of components of these vectors. Sections 4.09 and 4.10 describe some simulation tests, which show just how the system performance varies with these parameters, for two separate transmission channels. The results of these tests may be used to estimate the values of these parameters, needed to obtain the best performances that can be obtained with the systems.

The final section (4.11) of Chapter 4, is of interest when dealing with transmission channels whose sampled impulse response has some very small initial components. The simulation results in this section compare the performances of Systems 1-4 with modified versions of these systems, which ignore the first component of the channel'ssampled impulse response. These tests give a useful guide, when deciding whether a component of a certain size in the sampled impulse response, would be better ignored or taken into account, by the detector.

Chapter 5 is concerned with a phenomenon called merging which, when it occurs, can cause a sudden drop in the performances of Systems 1 and 2. This phenomenon is one in which several of the vectors stored in the detection process, become the same and remain locked in this state for long periods. This effectively reduces the number of possible data sequences which can be considered by the process, and can therefore lead to a loss of system performance. Section 5.02 gives a formal definition of the term, *"Merged vectors"*, and provides an upper bound for the probability of two vectors becoming merged.

It is shown in Section 5.04, that Systems 1 and 2 can sometimes get locked in a state, where the number of distinct vectors stored by the processes, is half of the total number of stored vectors. This state is referred to as, *"The failure mode"*. In Sections 5.05 - 5.09, the probability of System 1 eventually entering the failure mode, is estimated by means of two separate approaches. In the first, a theoretical model of the system is set up, which will give the probability of the failure mode occurring, if certain transition probabilities are known. These probabilities are estimated from the results of simulation tests. The second approach arrives at an estimate of the probability of System 1 entering the failure mode, by simulation testing alone, and without the use of the model. The results of the two approaches are compared in Section 5.07.

Section 5.10 discusses two modifications to System 1, which may be used to prevent the detection process from entering the failure mode. Evidence of the effectiveness of the first method is provided by means of simulation tests, carried out for a particular situation, in which System 1 suffered very noticeably from the effects of merging. The second method is proved generally effective at keeping the stored vectors distinct, by means of theoretical analysis.

1.05 Digital and Analogue Signals

An analogue signal may be defined to be one in which the signal waveform may take on an infinite number of possible shapes. When information (i.e. data) is transmitted in the form of analogue signals, the detected waveform at the receiver will usually be corrupted, to some extent, by noise. Even if the noise is of a fairly low intensity, the detected waveform may differ slightly from the waveform transmitted.

The situation may be improved by the use of digital signals, (i.e. signals whose waveform may take on one of a finite number of fairly distinct shapes). [7]. Then for a finite amount of data and a system which introduces no noise, there will be a finite set of possible received waveforms (or received signals), each one corresponding to some transmitted waveform. With a suitable detector, the appropriate transmitted signal can be derived from any member of this set of received signals.

For a practical system which does introduce noise, the received message will not be one of those in the set mentioned above. However, if the noise level is fairly low, the actual received waveform will be closely matched (in shape) to a member of this set, and the exact transmitted signal can be derived from this member.

If the data to be transmitted is in analogue form, it can be converted to a digital waveform without any loss of information [7]. From Nyquist's sampling theorem, this conversion to a digital waveform may be carried out by sampling the analogue waveform at regular intervals (i.e. every T seconds, for some value of T). If the sampling interval T is such that 1/(2T) is less than the highest frequency contained in the analogue waveform, these samples contain all of the relevant information, and the analogue signal can be recovered exactly from them [7]. This result allows the transmission of speech over a digital system without any information loss. If the noise level introduced by a digital data system is low, the speech waveform may be recovered at the receiver, as exactly as if there was no noise present.

1.06 Telephone Circuits [9]

Telephone circuits can generally be divided into two types: private lines and switched lines [9]. Switched lines are ones which are part of the public telephone network. They are made up from an almost random combination of different links and they generally cause more distortion than private lines.

Most telephone circuits consist of three different types of links called: unloaded links, loaded links and carrier links [9]. The unloaded links may consist of lengths of wire having an impedance of 600Ω and a length of about two or three miles. Being relatively short, the unloaded links have a good frequency response (i.e. their attenuation and delay distortions are moderate). The attenuation caused by the unloaded links is proportional to the square root of the frequency, over the voice frequency band (300 to 3000 Hz), and increases with distance at the rate of about 2½ dB per mile, in the centre of the band. This high increase in attenuation with length, prohibits the use of very long unloaded audio links. The delay distortion introduced by these links is negligible.

Loaded audio links may be much longer than the unloaded ones, with lengths up to about 100 miles. These links may consist of a pair of wires with inductances placed at regular intervals. (Typically 44 or 88 mH at lengths of 2000 yards). The loaded links have the same impedance (600Ω) as the unloaded ones, and have a frequency response similar to that of a low pass filter. Their attenuation per mile is less than 1 dB, up to a certain frequency, and then increases rapidly as the frequency rises. Hence the attenuation per mile of the loaded links, is considerably less than that of the unloaded links, at the centre of the voice frequency band. The delay distortion is about ten times as great as in the unloaded links and may be quite considerable. Loaded links require amplifiers at various stages along the lines, if they are to be more than a few miles in length. As amplifiers can only operate on a signal travelling in one direction, a separate pair of wires must then be used for transmission and reception.

Carrier links may be much longer than loaded audio links, and may consist of a coaxial cable or open wire lines, forming a wideband channel. With these links, the signal frequency band is shifted upwards by a linear process of amplitude modulation. Several signals are then transmitted simultaneously using an arrangement of frequency division multiplexing, each signal being sent on a separate frequency band. The distortion of a voice frequency signal transmitted over carrier links, is almost exclusively caused by the filters at each end, which are used for the linear modulation - demodulation process. The resulting frequency response is effectively that of a high pass filter, with attenuation rising rapidly below some cut off frequency (200 to 300 Hz). Delay distortion is considerable at frequencies just above this cut off frequency, and there is some attenuation at the high end of the voice frequency band.

Microwave satellite and PCM (pulse code modulation) links are also used in telephone circuits. These links are made to a high standard, and their attenuation and delay distortions are small in comparison to the more common types of lines in telephone circuits. If a data transmission system functions satisfactorily over loaded and unloaded audio links, and the poorer carrier links, it should not have any problems over satellite, microwave and PCM links.

1.07 Attenuation and Delay Distortions Over Telephone Circuits [9]

Figures 1.02 and 1.03 show the ideal attenuation-frequency and group delay-frequency characteristics for a telephone circuit. The group delay curve is flat over the voice frequency band, i.e. over the range 300 to 3000 Hz. The attenuation increases rapidly outside of the voice frequency band, so the behaviour of the group delay curve is not important there. The rapid increase of attenua-

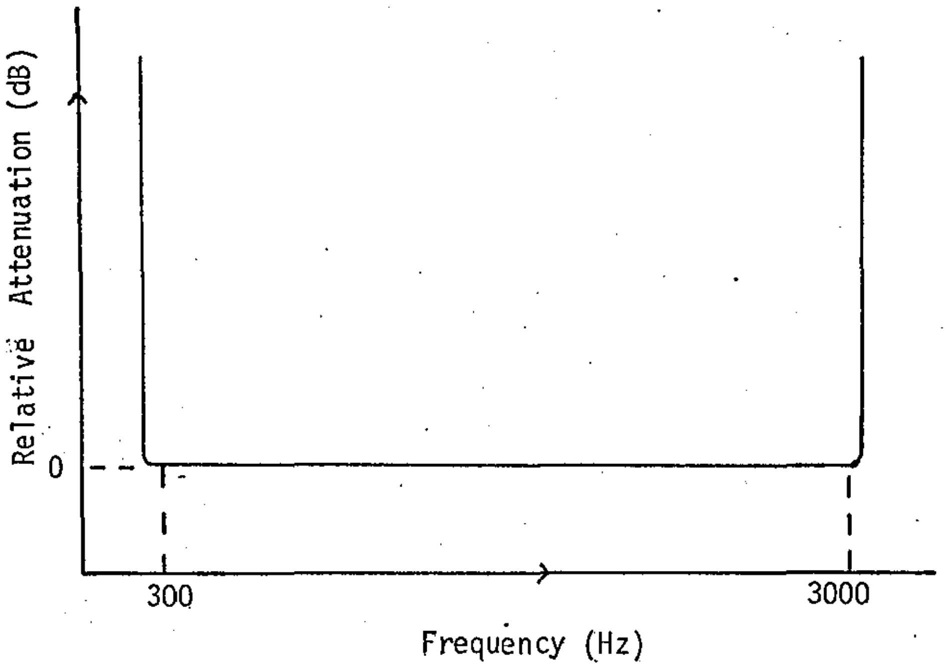


FIGURE 1.02

Attenuation-Frequency Characteristic for an Ideal Channel

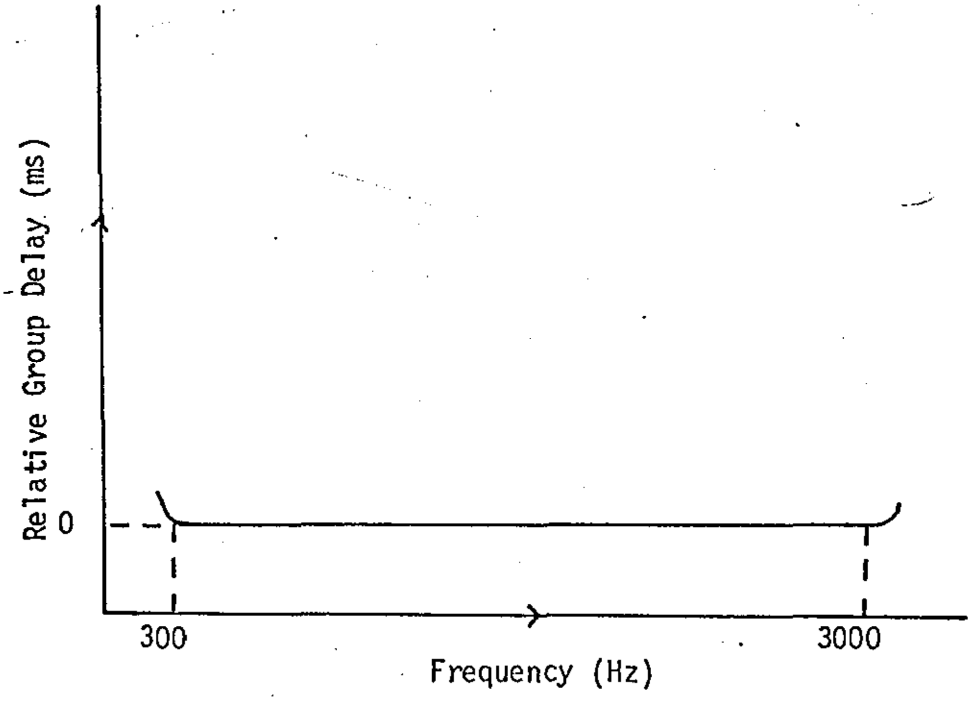


FIGURE 1.03

Group Delay-Frequency Characteristic for an Ideal Channel tion outside of the voice frequency band is desirable, so that unwanted signals with energy in frequencies outside of this band, may be eliminated to some extent.

Figures 1.04 and 1.05 show two typical characteristics, for telephone circuits containing both audio and carrier links. If a ten dB variation in attenuation can be tolerated, the whole of the voice frequency band is available for transmission, over the circuit represented by Figure 1.04.

Figures 1.06 and 1.07 show two characteristics of poor telephone circuits. For the circuit corresponding to Figure 1.06, the whole of the voice frequency band is not available, unless variations in attenuation of more than 20 dB can be tolerated. It should be noted that these characteristics vary greatly over different telephone circuits. A ripple is often present in both the attenuation and group delay characteristics.

Clearly, if the group delay-frequency characteristic is not flat over the voice frequency band, the received signal will be dispersed in time, with some frequency components arriving later than others. Hence unless the rate of transmission is kept below a certain level, the received signal elements corresponding to the different data elements, will overlap. Tests have shown that the time dispersion produced by telephone circuits does not usually exceed six milliseconds.

The attenuation at about 1000 Hz (this is usually the minimum level on the attenuation-frequency characteristic), may be as much as 30 dB for a switched line, but is usually less than 15 dB on a

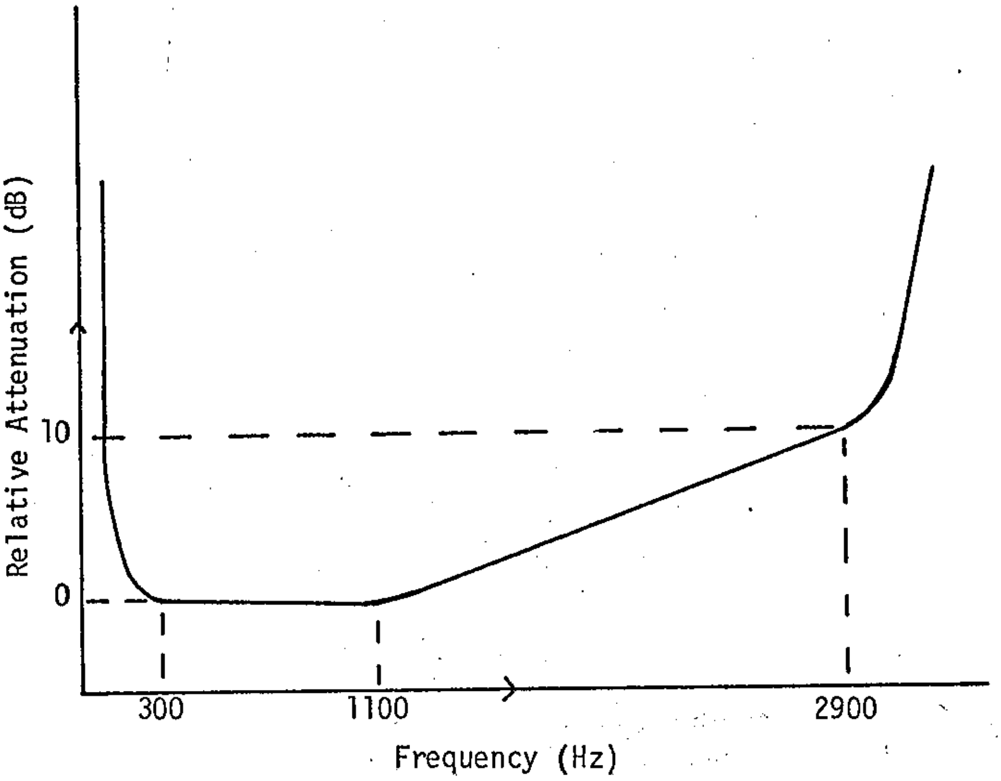


FIGURE 1.04

Attenuation-Frequency Characteristic for a Typical Channel

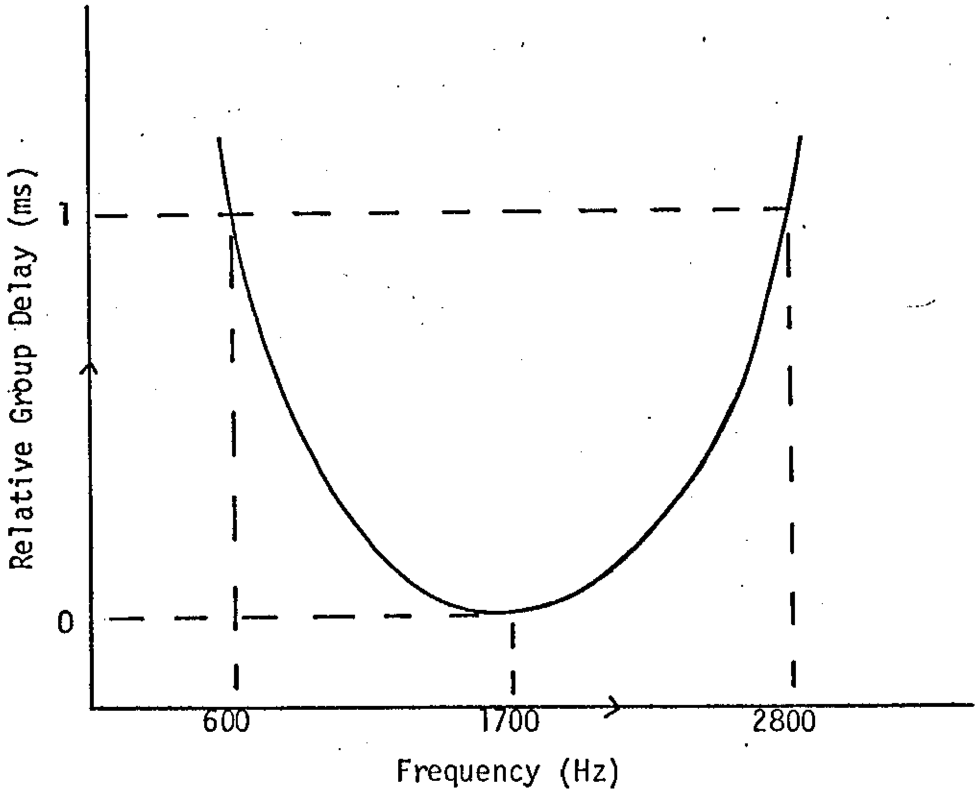


FIGURE 1.05

Group Delay-Frequency Characteristic for a Typical Channel

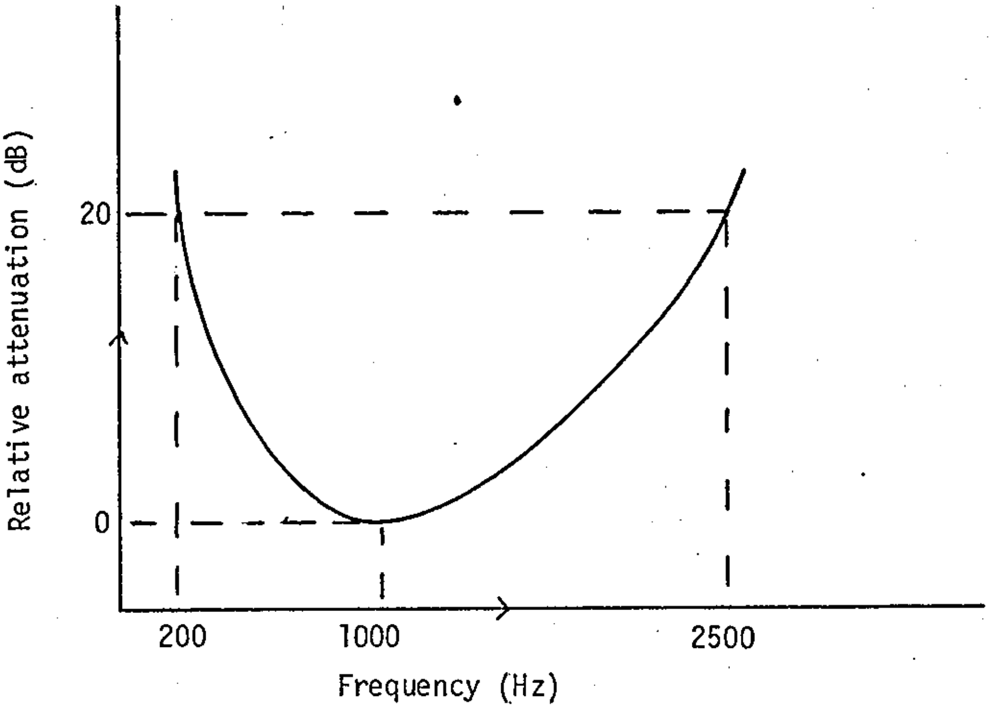


FIGURE 1.06

Attenuation-frequency characteristic for a poor channel

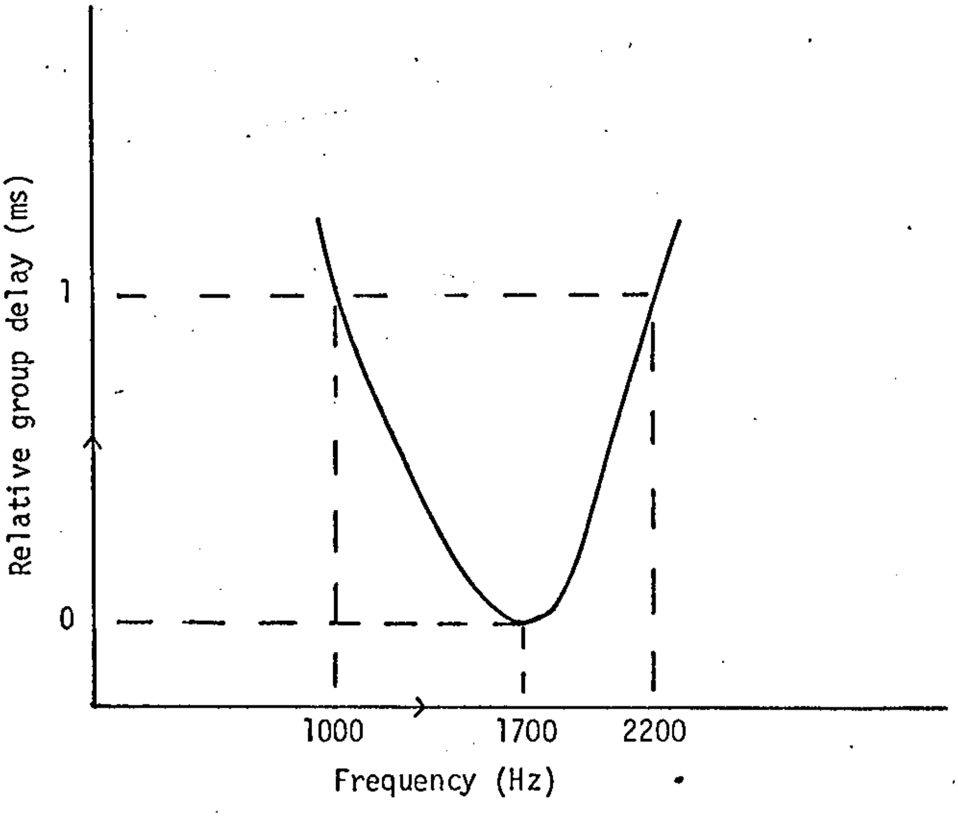


FIGURE 1.07

Group delay-frequency characteristic for a poor channel.

private line. Rising attenuation with frequencies above 1000 Hz is a common characteristic of telephone circuits.

1.08 H.F. Radio Links [9]

High frequency (H.F) radio links work on the same basic principle as carrier links, having a number of signals transmitted separately on different frequency bands. These frequency bands are contained within the range 3 to 30 MHz.

Whereas the distortion characteristics of most telephone circuits are fairly constant, this is not the case with H.F. radio links. The H.F. links suffer from an effect known as frequency selective fading, which causes a variation of the characteristics with time. This fading may occur if the transmitted signal takes more than one path from the transmitter to the receiver. The signal is then said to suffer from multipath propagation. One example of this phenomenon is the case where the radio waves are reflected from the ionosphere and the ground, perhaps several times. The waves reaching the receiver via different routes, typically have a difference in delay of about one or two milliseconds. The difference in delay, and the actual paths the signals take from the transmitter to the receiver, will of course vary with the height of the ionosphere. Hence the attenuation-frequency and group delay-frequency characteristic of H.F. radio links, may vary with time.

As with telephone circuits, the time dispersion of signals transmitted over H.F. radio links is usually less than six milliseconds. The dispersion with these links is, however, of a more harmful nature, as there may be more energy in the latter part of the dispersed signal, than is found with telephone circuits.

The changes in the distortion characteristics are periodic with cycles usually occurring at the rate of four to fifteen per minute. This rate at which the characteristics change is, in many cases, slow enough so that the receiver equipment can continually estimate them and adapt to the changes. The variation of the level in the received signal, due to fading, is typically up to 40 dB.

In telephone circuits, low cost is usually a priority, but, with H.F. radio links, more effort is made to ensure that the equipment is of a high standard. Hence the distortion occurring in the transmitted signal is due almost entirely to the transmission path, and not to the components of the data transmission system.

1.09 Random Noise [9]

The noise appearing in telephone circuits can be divided into two categories: additive noise and multiplicative noise. The additive noise takes the form of a random signal added to the transmitted signal, whereas multiplicative noise modulates the signal waveform. When the noise is sufficiently intense, the received signal waveform may be mistaken for one corresponding to the wrong transmitted signal and errors may occur in the detection process. If the main cause of errors is additive noise, the error rate may be reduced by increasing the signal level. For the case of multiplicative noise, the noise level in the received signal is proportional to the level of the transmitted signal, so the error rate cannot be reduced in this way.

White Gaussian Noise (WGN) is a waveform with a constant two sided power spectral density. It has the property that the value of its waveform at any time, is a normally distributed random variable with zero mean. Two samples of the waveform taken at any distinct times are also statistically independent. This type of noise is not physically realisable, but it can be modelled by a real waveform with a power spectral density function which is constant over a wide range of frequencies.

Additive WGN is not a type of noise that occurs with great intensity over telephone circuits. However tests and theoretical considerations have shown that systems which have a good tolerance to this type of noise, also have a good tolerance to other forms of additive noise [6]. Usually, if one system has a better tolerance to this noise than another system, it will also have a better tolerance to other additive noise.

Additive WGN is relatively easy to simulate, is easy to work with in practice, and is often the only type of noise used in testing data transmission systems.

Over switched telephone circuits, the majority of noise is additive, but multiplicative noise is more common over private lines. The tests and theoretical analysis in this thesis assume that additive WGN is the only type of noise present in the systems. Strictly speaking, systems which perform well under these conditions, will not have been shown to be the best for use on private lines.

Over H.F. radio links, the main source of additive noise is atmospheric noise caused by lightening. It can be shown that Gaussian noise is a reasonable model for atmospheric noise. As before, tolerance of a system to additive WGN, is a good guide to the tolerance to the various types of additive noise present.

1.10 The Linear Equalizer [6, 14-18]

The linear equalizer (or linear transversal filter), is among the simplest of detection processes for signals with intersymbol interference. It consists of a network of delays and multipliers, as shown in Figure 1.08.

The samples $r_0, r_1, \ldots, r_n$ are fed to the input of the filter at intervals of T seconds, say, and the delays are the same length as these intervals. The delays are such that, if they receive a sample value r_i at their input at time iT, this value r_i will appear at their output at time $(i + 1)T$. The multipliers with coefficients y_i, produce an output equal to y_i multiplied by their input. The outputs from the multipliers are added together, so that the output of the equalizer at time iT is given by:

$$t_i = \sum_{h=0}^{f} r_{i-h}\, y_h \qquad \text{for} \qquad i = 0, 1, \ldots, n + f \tag{1.03}$$

where $r_i \overset{\Delta}{=} 0$ for i not contained in the set $\{0, 1, \ldots, n\}$.

The z transform of a sequence of numbers $s_0, s_1, \ldots, s_n$ is defined by

$$f(z) = s_0 + s_1 z^{-1} + \ldots. + s_n z^{-n} \tag{1.04}$$

Let R(z) and Y(z) be the z transforms of the input sequence $\{r_i\}$ and the sequence of multiplier coefficients $\{y_i\}$, respectively.

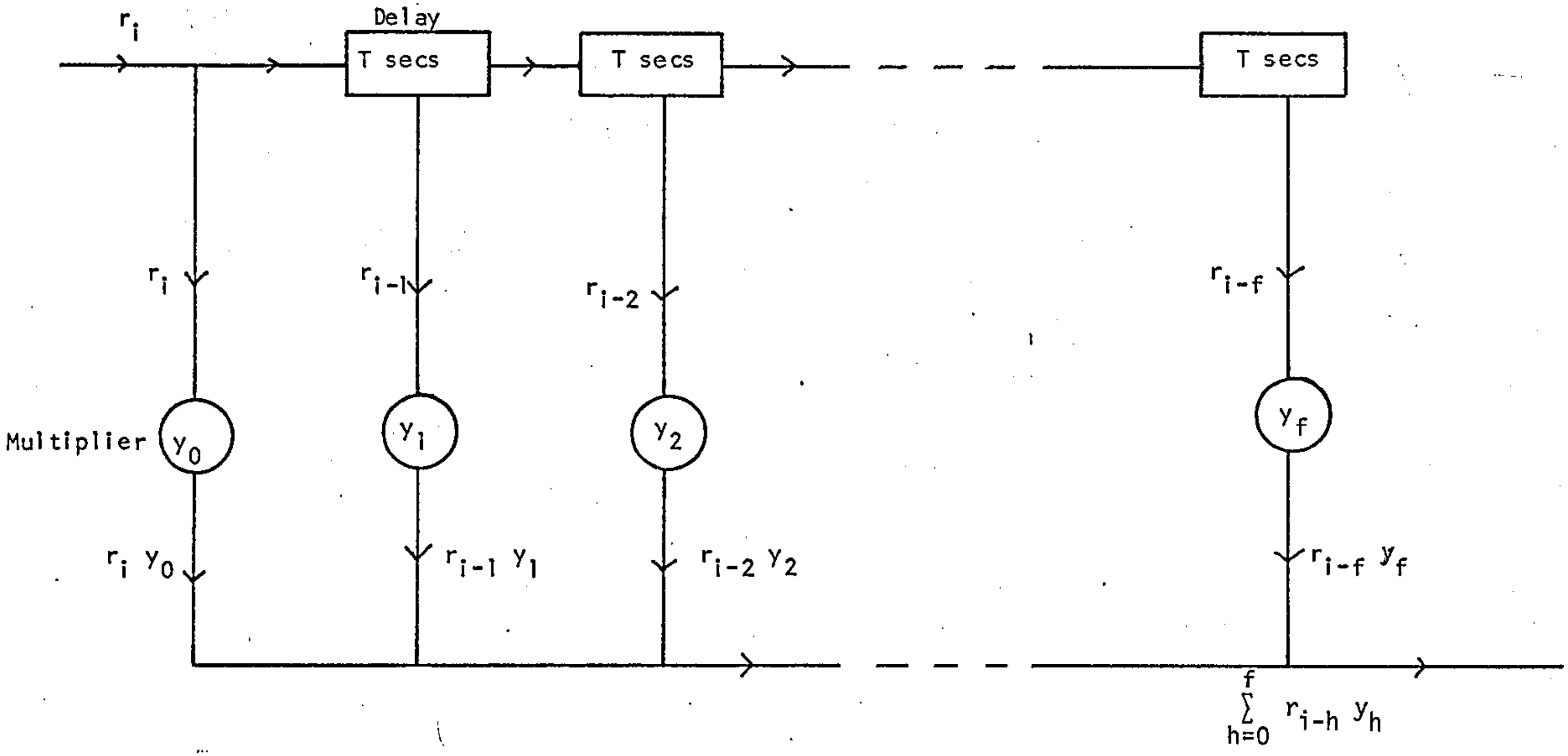


FIGURE 1.08 A Linear Equalizer

Then

$$R(z) = \sum_{h=0}^{n} r_h \, z^{-h} \quad \text{and} \tag{1.05}$$

$$Y(z) = \sum_{i=0}^{f} y_i \, z^{-i} \tag{1.06}$$

Hence:

$$R(z)\, Y(z) = (y_0 + y_1 \, z^{-1} + \ldots + y_f \, z^{-f})(r_0 + r_1 \, z^{-1} + \ldots + r_n \, z^{-n})$$

$$= y_0 \, r_0 + z^{-1} \, (y_0 \, r_1 + y_1 \, r_0) + z^{-2} \, (y_0 \, r_2 + y_1 \, r_1 + y_2 \, r_0)$$
$$+ \ldots$$

$$= z^0 \sum_{i+h=0} y_i \, r_h + z^{-1} \sum_{i+h=1} y_i \, r_h + z^{-2} \sum_{i+h=2} y_i \, r_h + \ldots$$

where i and h are restricted to being ≥ 0.

$$R(z)\, Y(z) = z^0 \sum_{i=0}^{f} y_i \, r_{-i} + z^{-1} \sum_{i=0}^{f} y_i \, r_{1-i} + z^{-2} \sum_{i=0}^{f} y_i \, r_{2-i} + \ldots$$

$$\ldots + z^{(-n-f)} \sum_{i=0}^{f} y_i \, r_{n+f-i} \tag{1.07}$$

where $r_i \overset{\Delta}{=} 0$ for i not contained in the set $\{0, 1, \ldots, n\}$.

The output of the equalizer at time iT is given by:

$$t_i = \sum_{h=0}^{f} r_{i-h} \, y_h \qquad \text{for} \quad i = 0, 1, \ldots, n + f$$

(see equation 1.03). Hence the z transform of the output of the equalizer is

$$z^{0}\sum_{h=0}^{f} r_{-h}\, y_h + z^{-1}\sum_{h=0}^{f} y_h\, r_{1-h} + \ldots + z^{-(n+f)}\sum_{h=0}^{f} y_h\, r_{n+f-h}$$

which is equal to R(z) Y(z) (see equation 1.07). Hence the z transform of the output sequence from the equalizer, is equal to the product of the transforms, of the input sequence and the sequence of multiplier coefficients. The sequence $y_0, y_1, \ldots, y_f$ is called the sampled impulse response of the equalizer.

A similar result with z transforms applies to the case where the sequence $\{s_i\}$ is used to modulate impulses, which are then sent over a transmission channel, as in the model described in Section 1.02. From equation 1.02, the output from the baseband channel at time t is given by

$$r(t) = \sum_{i=0}^{n} s_i\, y(t - iT) + w(t)$$

where y(t) is the impulse response of the channel and w(t) is a function representing random noise. r(t) is sampled at intervals of T seconds to give a sequence $\{r_i\}$ of received signal samples, which is fed to the detector. Suppose that the first sample is taken at time 0 so that the received samples are given by

$$r(jT) = \sum_{i=0}^{n} s_i\, y(jT - iT) + w(jT) \qquad (1.08)$$

Let p and q be the smallest and largest integers respectively, such that $y(pT) \neq 0$ and $y(qT) \neq 0$. Let $g = q - p$ and $y_i = y[(p+i)T]$ for $i = 0, 1, \ldots, g$. Then the vector

$$(y_0, y_1, \ldots y_g)$$

is called the sampled impulse response of the transmission channel.

Now let

$$r_k = r\,[(p + k)\,T] \quad \text{and}$$

$$w_k = w\,[(p + k)\,T].$$

Then, from equation 1.08,

$$r_k = \sum_{i=0}^{n} s_i\; y\,[(p + k - i)\,T] \; + w_k$$

But $y[(p + k - i)\,T] = 0$ for $k - i < 0$ or $k - i > g$ (this follows from the definitions of p and q).

$$\therefore \quad r_k = \sum_{i=k}^{k-g} s_i\; y\,[(p + k - i)\,T] \; + w_k$$

$$r_k = \sum_{i=0}^{g} s_{k-i}\, y_i + w_k \qquad (1.09)$$

for $k = 0, 1, 2, \ldots.$

Let the z transforms of the sequences $\{r_i\}$, $\{s_i\}$, $\{y_i\}$ and $\{w_i\}$ be R(z), S(z), Y(z) and W(z) respectively. During the analysis of the linear equalizer, it was shown that the z transform of the sequence whose i th term is

$$\sum_{h=0}^{g} r_{i-h}\, y_h \; ,$$

is the product of R(z) and Y(z).

Hence the z transform of the sequence whose k th term is

$$\sum_{h=0}^{g} s_{k-h}\, y_h$$

is S(z) Y(z) and, from equation 1.09,

$$R(z) = S(z)\ Y(z) + W(z)$$

Now let the sequence r_k, of received signal samples, be fed to a linear equalizer whose z transform is Y*(z). Then the z transform of the output of the equalizer is given by

$$R^*(z) = R(z)\ Y^*(z)$$

$$= [S(z)\ Y(z) + W(z)]\ Y^*(z)$$

$$R^*(z) = S(z)\ Y(z)\ Y^*(z) + W(z)\ Y^*(z) \qquad (1.10)$$

It is often possible to choose the multiplier coefficients of the equalizer to give a z transform Y*(z) such that

$$Y^*(z)\ Y(z) \simeq z^{-k} \qquad (1.11)$$

for some integer $k \geq 0$. Then the z transform of the sampled impulse response, of the combined channel and equalizer, is approximately of the form

$$(0, 0, \ldots, 0, 1, 0, \ldots, 0).$$

Thus the combination of the channel and equalizer only introduces a delay in the data sequence $\{s_i\}$, and there is no signal distortion.

Then, from equation 1.10, the output of the equalizer has z transform

$$R^*(z) = S(z)\ z^{-k} + W(z)\ Y^*(z) \tag{1.12}$$

Now let

$$W(z)\ Y^*(z) = u_0 + u_1\ z^{-1} + u_2\ z^{-2} + \ldots \tag{1.13}$$

for some set of coefficients u_0, u_1, u_2, ... and let the samples at the output of the equalizer.be r_0^*, r_1^*, r_2^*, ..., so that

$$R^*(z) = r_0^* + r_1^*\ z^{-1} + r_2^*\ z^{-2} + \ldots \tag{1.14}$$

Then from equations 1.12, 1.13 and 1.14

$$r_0^* + r_1^*\ z^{-1} + r_2^*\ z^{-2} + \ldots = (s_0 + s_1\ z^{-1} + s_2\ z^{-2} + \ldots)z^{-k}$$

$$+ u_0 + u_1\ z^{-1} + u_2\ z^{-2} + \ldots$$

Hence, equating coefficients of z^{i+k},

$$r_{i+k} = s_i + u_{i+k} \tag{1.15}$$

Now, from equation 1.13, it can be seen that the sequence $\{u_i\}$ is formed from the convolution of the sequence $\{w_i\}$, and the sequence $\{y_i^*\}$ which forms the z transform $Y^*(z)$. Hence each term u_i is a linear combination of the terms from the sequence $\{w_i\}$, which are samples from a white Gaussian waveform with zero mean. It therefore follows that each u_i must have zero mean. Hence, from equation 1.15,

$$r_{i+k} = s_i + u_{i+k}$$

where u_{i+k} is a random variable with zero mean. Each element s_i is then detected as the possible data element value which is closest to r_{i+k}.

The random variables u_i are formed from a linear combination of the independent random variables w_i, so the u_i terms will not be independent of each other. This fact is of no disadvantage if the linear equalizer is used as a detector, in the manner described above. The linear equalizer is, however, sometimes used in conjunction with other detection processes, as mentioned in Section 1.03. When this equalizer is used with a V.A. detector, it is usually placed between the detector and the transmission path. Then the equalizer's coefficients are chosen in such a way, that the combination of channel and equalizer has a shorter impulse response than that of the channel alone, (i.e. the sampled impulse response of the combination has fewer components than that of the channel). As far as the V.A. detector is concerned, the original channel has then been replaced by one with fewer componentsin its sampled impulse response. In Chapter 2, it is shown that the amount of computation required by the V.A., increases rapidly with the number of components of the channel's sampled impulse response. It may therefore be seen that the use of a linear filter, with the V.A. detector, will allow a reduction in computation.

The combination of linear filter and V.A. detector, does have the drawback that the linear filter usually causes a correlation of the noise samples (i.e. noise samples which are independent, at the input to the filter, will give rise to noise samples which are not independent at the filter's output). This correlation effect may cause some loss in the tolerance to additive white Gaussian noise, of the detection process.

It may be shown that a linear filter which causes only pure phase distortion (see Section 1.11), does not cause a correlation of the noise samples [6]. It may not, however, be very beneficial to use this type of filter with the V.A. detector, if it cannot effectively shorten the sampled impulse response of the transmission channel, to any great extent. This type of pure phase equalizer can be used with advantage though, with some of the detection processes described in Chapter 3.

1.11 Phase Distortion and Amplitude Distortion

Consider a linear filter with sampled impulse response (y_0, y_1, ..., y_g). The z transform of the sequence y_0, y_1, ..., y_g is defined by

$$Y(z) = y_0 + y_1 z^{-1} + \dots + y_g z^{-g}.$$

The reverse of this z transform is given by

$$X(z) = y_g + y_{g-1} z^{-1} + \dots + y_0 z^{-g}$$

A filter causing pure phase distortion may be defined as one whose sampled impulse response $(y_0, y_1, \dots, y_g)$ is such that

$$Y(z) X(z) \approx z^{-k}$$

for some integer $k \geq 0$. Hence, the z transform of the filter formed from the filter Y(z) in series with X(z), is z^{-k}. Hence the combination of the two filters has a sampled impulse response of the form (0, 0, 1, 0, - , 0), and no distortion is caused by the combined filter.. i.e. apart from the delay introduced by the combined channel,

$$X(z) = [Y(z)]^{-1}$$

Therefore a channel causing pure phase distortion may be defined as one whose z transform Y(z), is such that its reverse X(z), is also its inverse (neglecting the delay represented by the term z^{-k}).

It is not possible, in fact, for the equation

$$Y(z) X(z) = z^{-k}$$

to hold, for any finite sequence $y_0, y_1, \dots, y_g$. So, strictly speaking, pure phase distortion is not possible with a filter (or transmission channel) whose sampled impulse response has a finite number of components. However, if the number of terms is fairly large, it is possible to get close to the case of pure phase distortion.

A filter introducing pure amplitude distortion, may be defined as one whose sampled impulse response $(y_0, y_1, \ldots, y_g)$ has an odd number of terms, and is symmetric in the sense:

$$y_0 = y_g$$

$$y_1 = y_{g-1}$$

$$\vdots$$

$$y_{\frac{1}{2}g-1} = y_{\frac{1}{2}g+1} \quad .$$

A typical filter or transmission channel, will introduce both amplitude and phase distortion. It is not usually a straightforward matter to determine the degree of each of these types of distortion, from a given sampled impulse response.

It is possible to assess the degree of amplitude distortion, to some extent, as follows.

Let the sampled impulse response of the channel or filter under consideration be $(y_0, y_1, \ldots, y_g)$, as before. Let

$$b_i = \sum_{h=0}^{g} y_h \, y_{i-g+h} \qquad \text{(where } y_i \overset{\Delta}{=} 0 \text{ for i not contained in the set } \{0, 1, \ldots, g\})$$

for i = 0, 1, ..., g and let

$$d = \frac{1}{b_g} \sum_{i=0}^{g-1} |b_i| \qquad (1.16)$$

Then it may be shown [6] that the magnitude of d gives a measure of the degree of amplitude distortion, i.e. a large value of d indicates that the amplitude distortion is severe. It will be seen in Chapter 4, that the channels with the largest values of d, usually give the poorest tolerance to additive white Gaussian noise.

1.12 The Feedback Transversal Equalizer Using Decision Directed Cancellation [6, 14, 20-24]

Consider a baseband channel with sampled impulse response $(y_0, y_1, \ldots, y_g)$, in the data transmission system described in Section 1.02. The sampled received signal will form a sequence $\{r_i\}$ such that

$$r_i = y_0 s_i + y_1 s_{i-1} + \ldots + y_g s_{i-g} + w_i$$

(see equation 1.09), where $s_0, s_1, \ldots, s_n$ is the data sequence and $\{w_i\}$ is a sequence of noise samples. The detector described below works reasonably well, if one of the components y_i is large in comparison to the others. Suppose that y_j is such a component, for some integer j such that $0 \leq j \leq g$. A training signal of known elements s_i is sent out by the transmitter, before the actual data sequence, so that the detector has knowledge of the recent elements transmitted. Each data element s_i is then detected from the received signal sample r_{i+j}, as follows:

From equation 1.09,

$$r_{i+j} = y_0\, s_{i+j} + y_1\, s_{i+j-1} + \ldots + y_j\, s_i$$

$$+ y_{j+1}\, s_{i-1} + \ldots + y_g\, s_{i+j-g} + w_{i+j}$$

By the time r_{i+j} arrives at the detector, $s_{i-1}, s_{i-2}, \ldots, s_{i+j-g}$ will have been detected. Hence the expression

$$t_{i+j} = y_{j+1}\, s_{i-1} + y_{j+2}\, s_{i-2} + \ldots + y_g\, s_{i+j-g}$$

can be calculated, assuming that $s_{i-1}, s_{i-2}, \ldots, s_{i+j-g}$ have been detected correctly. Now let

$$R_{i+j} = r_{i+j} - t_{i+j} \quad .$$

Then

$$R_{i+j} = y_0\, s_{i+j} + y_1\, s_{i+j-1} + \ldots + y_j\, s_i + w_{i+j}$$

$$\therefore \quad s_i = \frac{R_{i+j}}{y_j} - \frac{(y_0\, s_{i+j} + y_1\, s_{i+j-1} + \ldots + y_{j-1}\, s_{i+1} + w_{i+j})}{y_j} \qquad (1.17)$$

Now, if y_j is much larger than $y_0, y_1, \ldots, y_{j-1}$,

$$\frac{y_0\, s_{i+1} + y_1\, s_{i+j-1} + \ldots + y_{j-1}\, s_{i+1}}{y_j}$$

will be small and

$$s_i \simeq \frac{R_{i+j}}{y_j} - \frac{w_{i+j}}{y_j}$$

s_i is then detected as the possible value of a data element which is closest to R_{i+j}/y_j.

If the first channel component y_0 is reasonably large, it is usually best to take $j = 0$, so that each data element s_i is detected from r_i. Then

$$t_{i+j} = t_i$$

$$= y_1 \; s_{i-1} + y_2 \; s_{i-2} + \ldots + y_g \; s_{i-g}$$

and

$$R_{i+j} = R_i$$

$$= r_i - t_i$$

$$= y_0 \; s_i + w_i$$

Then R_i depends only on s_i and not on s_{i-1}, s_{i-2}, ..., s_{i-g}, so that the contribution to R_i from all data elements other than s_i, has been removed before s_i is detected. If y_0 is small, this arrangement does not perform very well, as the value of R_i will then be influenced more by the noise component w_i than by $y_0 \; s_i$. The expression for t_{i+j} is evaluated by means of a linear transversal filter and is subtracted from r_{i+j}, as shown in Figure 1.09. The decision mechanism then selects the possible value of a data element which is closest to R_{i+j}/y_j and assigns this value to s_i.

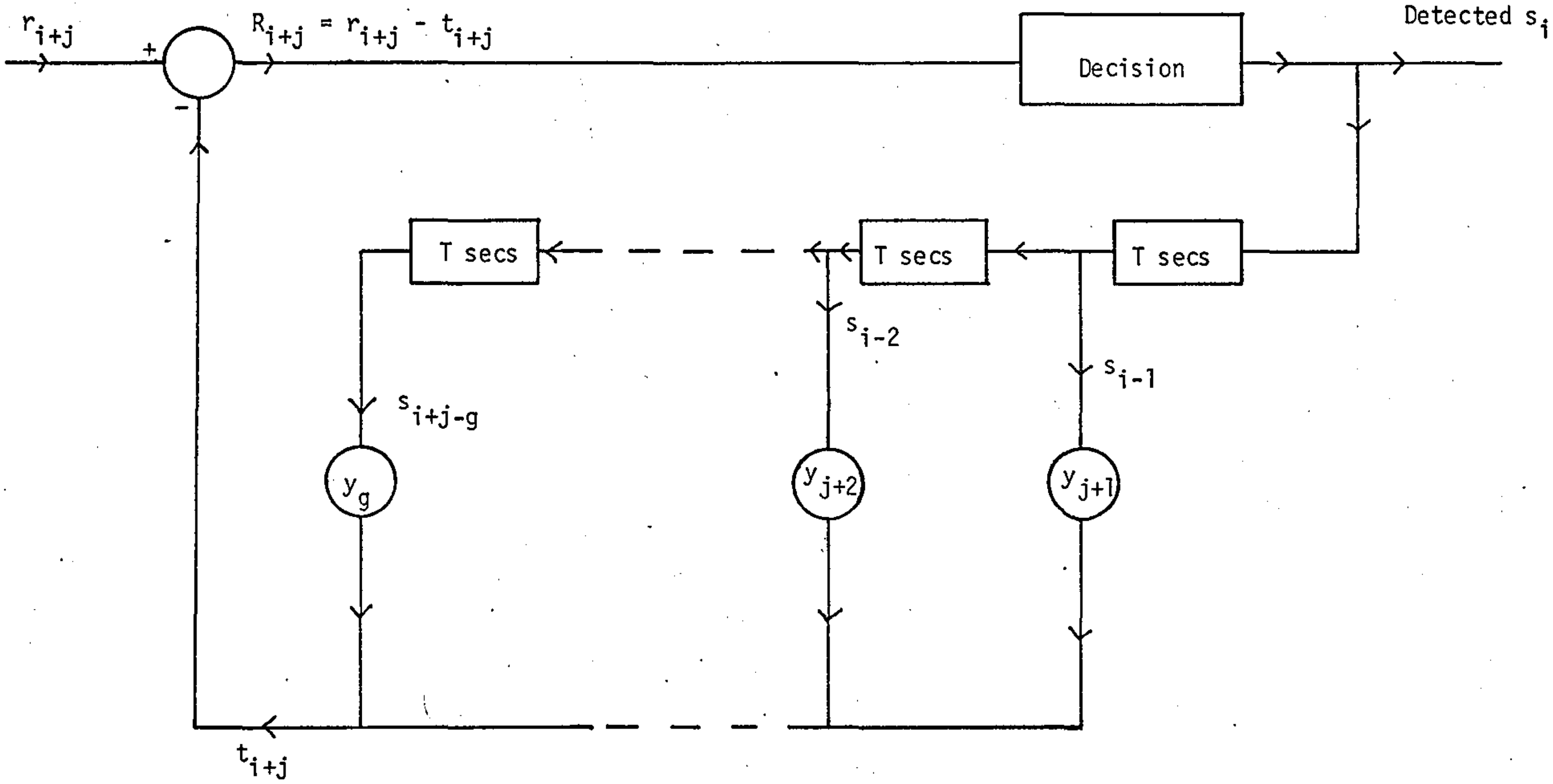


FIGURE 1.09 A nonlinear equalizer using a feedback transversal filter

1.13 A Combined Decision Feedback and Linear Equalizer [6, 14, 20-24]

This is a detector which makes use of both of the processes described in Sections 1.10 and 1.12. Consider the case where the feedback transversal equalizer of Section 1.12, is used with j chosen such that y_j is the largest component of the sampled impulse response $(y_0, y_1, \ldots, y_g)$, of the channel. From equation 1.17,

$$s_i = \frac{R_{i+j}}{y_j} - \frac{(y_0\, s_{i+j} + y_1\, s_{i+j-1} + \ldots + y_{j-1}\, s_{i+1} + w_{i+j})}{y_j}$$

In this expression for s_i, the terms containing $y_{j+1}, y_{j+2}, \ldots, y_g$ are not present, as these have been removed from r_{i+j}, by means of the feedback equalizer. s_i is then detected as the possible value of a data element which lies closest to R_{i+j}/y_j. If the terms $y_0, y_1, \ldots, y_{j-1}$ are significantly large, they will make up a sizeable contribution to the right hand side of equation 1.17. It will then no longer be true that

$$s_i \simeq \frac{R_{i+j}}{y_j} - \frac{w_{i+j}}{y_j}$$

In this case, the performance of the decision feedback equalizer may not be satisfactory. However in many cases it is possible to use a linear feed forward transversal filter, to effectively remove the terms containing $y_0, y_1, \ldots, y_{j-1}$, from the right hand side of equation 1.17, i.e. it may be possible to choose the linear filter in such a way, that the sampled impulse response of the channel in series with the linear filter, is $(\alpha y_j, \alpha y_{j+1}, \ldots, \alpha y_g)$ for some constant α. [6, 14]. A diagram of a detector using this type of linear filter, and a decision feedback equalizer, is given in Figure 1.10. With the terms involving $y_0, y_1, \ldots, y_{j-1}$ removed, equation 1.17 becomes

$$s_i = \frac{R_{i+j}}{y_j} - \frac{w_{i+j}}{y_j}$$

and the data sequence $\{s_i\}$ may be detected more accurately.

There is in fact, great freedom of choice in the particular combination of linear filter and feedback transversal filter used to equalize the channel. It is therefore possible to choose an arrangement which maximises the tolerance of the system to additive white Gaussian noise [6, 14].

Let the z transform of the channel's sampled impulse response be given by

$$A(z) = (z^{-1} - \alpha_1)\ (z^{-1} - \alpha_2)\ \ldots\ (z^{-1} - \alpha_p)$$

for some integer p, where α_1^{-1}, α_2^{-1}, ..., α_p^{-1} are the roots of A(z). Let β_1^{-1}, β_2^{-1}, ..., β_q^{-1} be the roots which satisfy the condition $|\beta_i^{-1}| > 1$, where $0 \leq q \leq p$. Then it may be shown, that the combination of equalizers which gives the greatest tolerance to additive white Gaussian noise, is the one in which the linear filter has the z transform

$$\frac{(\overline{\beta}_1 z^{-1} - 1)\ (\overline{\beta}_2 z^{-1} - 1)\ \ldots\ (\overline{\beta}_q z^{-1} - 1)}{(z^{-1} - \beta_1)\ \ (z^{-1} - \beta_2)\ \ \ldots.\ \ (z^{-1} - \beta_q)}$$

where $\overline{\beta}_i$ is the complex conjugate of β_i. [6, 14].

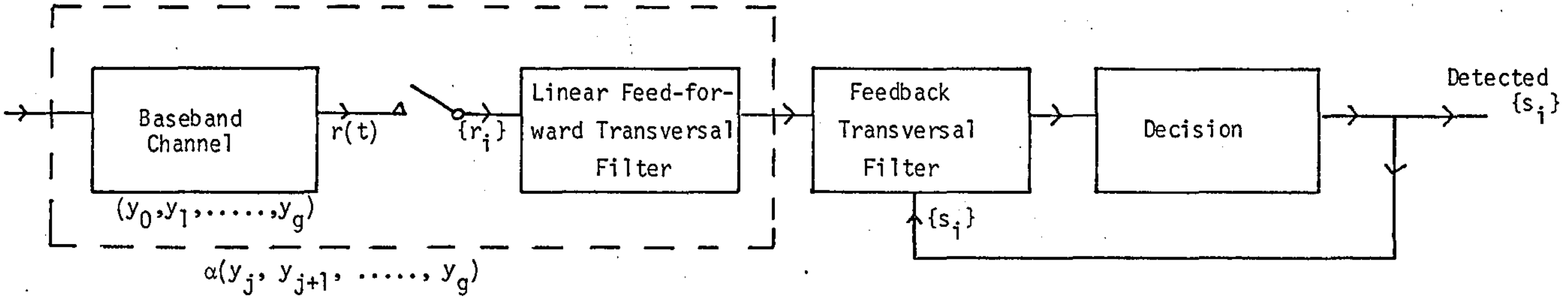

Baseband
Channel
r(t)
$\{r_i\}$
Linear Feed-forward Transversal Filter
$(y_0, y_1, \ldots\ldots, y_g)$
$\alpha(y_j, y_{j+1}, \ldots\ldots, y_g)$
Feedback Transversal Filter
$\{s_i\}$
Decision
Detected $\{s_i\}$

FIGURE 1.10 A combination of linear transversal filter and feedback transversal filter

1.14 Probability of Error for the Combined Linear and Decision Feedback Equalizer, when used with the Ideal Channel

Consider the optimum combination of equalizers described in Section 1.13, when used in conjunction with the ideal channel. (A channel with unity as the only non zero component of its impulse response). With this channel, each data element s_i is detected according to the position of the corresponding received sample r_i, relative to a set of decision thresholds. The detected value of s_i is the data element value which is closest to r_i.

First consider the case of a binary signal, so that the possible values of each data element s_i, are ±1. From equation 1.09,

$$r_i = s_i + w_i \quad .$$

(taking $y_0 = 1$ and $y_i = 0$ for $i \neq 0$) where w_i is a normally distributed random variable, with zero mean and variance σ^2.

Suppose that $s_i = 1$. Then

$$r_i = 1 + w_i$$

s_i will be detected as a 1 or -1, according to whether r_i is closer to 1 or -1, respectively. Hence s_i will be detected incorrectly if $w_i < -1$. Similarly when $s_i = -1$, it will be detected incorrectly if $w_i > 1$. Let

$$P_e = \text{Probability } (s_i \text{ is detected incorrectly}).$$

Then:

$$P_e = \text{Prob } (s_i = 1 \text{ and } w_i < -1 \text{ or } s_i = -1 \text{ and } w_i > 1)$$

$$= \text{Prob } (s_i = 1 \text{ and } w_i < -1) + \text{Prob } (s_i = -1 \text{ and } w_i > 1)$$

as the events $s_i = 1$ and $s_i = -1$ are mutually exclusive.

Therefore

$$P_e = \text{Prob } (s_i = 1) \text{ Prob } (w_i < -1) + \text{Prob } (s_i = -1) \text{ Prob}(w_i > 1)$$

The possible values of s_i are assumed to be equally likely (see Section 1.02) so

$$\text{Prob } (s_i = 1) = \text{Prob } (s_i = -1) = \tfrac{1}{2}$$

Hence

$$P_e = \tfrac{1}{2} \text{ Prob } (w_i < -1) + \tfrac{1}{2} \text{ Prob } (w_i > 1)$$

w_i is normally distributed with zero mean so, from the symmetry of the distribution,

$$\text{Prob } (w_i < -1) = \text{Prob } (w_i > 1)$$

$$\therefore \quad P_e = \tfrac{1}{2} \text{ Prob } (w_i > 1) + \tfrac{1}{2} \text{ Prob } (w_i > 1)$$

$$= \text{Prob } (w_i > 1).$$

If the variance of w_i is specified, Prob $(w_i > 1)$ can be found from tables of the normal probability distribution, and P_e can be evaluated.

Now consider the case of a quaternary signal, so that the possible data element values are ±1 and ±3. From equation 1.09,

$$r_i = s_i + w_i$$

s_i is detected as the data element value which is closest to r_i. For a case when $s_i = 3$, s_i will be detected incorrectly if r_i is closer to one of the values 1, -1, -3, than it is to 3. Hence s_i will be detected incorrectly if

$$3 + w_i < 2$$

or

$$w_i < -1$$

Similarly, if $s_i = -3$, it will be detected incorrectly if $w_i > 1$.

Now consider a case where $s_i = 1$. Then

$$r_i = 1 + w_i$$

and an error will occur if r_i is closer to one of the values 3, -1, -3, than it is to 1. Hence s_i will be wrongly detected if $|w_i| > 1$. Similarly, when $s_i = -1$, there will be an error if $|w_i| > 1$.

Hence, for the quaternary signal,

$$P_e = \text{Prob } [(s_i = 3 \text{ and } w_i < -1) \text{ or } (s_i = 1 \text{ and } |w_i| > 1)$$

$$\text{or } (s_i = -1 \text{ and } |w_i| > 1) \text{ or } (s_i = -3 \text{ and } w_i > 1)]$$

$$= \text{Prob}\ (s_i = 3)\ \text{Prob}\ (w_i < -1) + \text{Prob}\ (s_i = 1)\ \text{Prob}\ (|w_i| > 1)$$

$$+ \text{Prob}\ (s_i = -1)\text{Prob}\ (|w_i| > 1) + \text{Prob}\ (s_i = -3)\ \text{Prob}\ (w_i > 1)$$

But

$$\text{Prob}\ (|w_i| > 1) = \text{Prob}\ (w_i < -1 \text{ or } w_i > 1)$$

$$= \text{Prob}\ (w_i < -1) + \text{Prob}\ (w_i > 1)$$

$$= 2\ \text{Prob}\ (w_i > 1)$$

as the distribution is symmetric about zero. Therefore

$$P_e = \text{Prob}\ (s_i = 3)\ \text{Prob}\ (w_i > 1) + 2\ \text{Prob}\ (s_i = 1)\ \text{Prob}\ (w_i > 1)$$

$$+ 2\ \text{Prob}(s_i = -1)\ \text{Prob}\ (w_i > 1) + \text{Prob}\ (s_i = 3)\ \text{Prob}\ (w_i > 1)$$

The four possible values of a data element are equally likely, and occur with probability $\frac{1}{4}$. Therefore

$$P_e = \text{Prob}\ (w_i > 1)\ (\tfrac{1}{4} + 2 \times \tfrac{1}{4} + 2 \times \tfrac{1}{4} + \tfrac{1}{4})$$

$$= \frac{3}{2}\ \text{Prob}\ (w_i > 1).$$

As for the case of a binary signal, this probability of error, may be evaluated if the variance σ^2 is specified.

1.15 A Detection Process which deals with Data Transmitted in Distinct Groups

Consider once again the data transmission system described in Section 1.02, but with the modification that the data is sent in a number of distinct sequences, of the form $s_0, s_1, \ldots., s_n$. The corresponding sampled received signal is given by

$$r_i = \sum_{h=0}^{g} y_h \, s_{i-h} + w_i$$

for i = 0, 1, ..., n + g, (see Equation 1.09).

s_i is defined to be zero for $i < 0$ or $i = n + 1, n + 2, \ldots, n + g$. Thus there is a gap in transmission, of at least g elements, after the sequence $s_0, s_1, \ldots, s_n$ is transmitted. $(y_0, y_1, \ldots, y_g)$ is, of course, the sampled impulse response of the channel.

Now let

$$\underline{R} = (r_0, r_1, \ldots, r_{n+g})$$

$$\underline{W} = (w_0, w_1, \ldots, w_{n+g}) \text{ and}$$

$$\underline{S} = (s_0, s_1, \ldots, s_n).$$

Let Y be the (n+1) x (n + g+1) matrix given by

$$Y = \begin{vmatrix} y_0, & y_1, & \ldots, & y_g, & 0, & 0, & \ldots\ldots & 0 \\ 0, & y_0, & y_1, & \ldots, & y_g, & 0, & 0, & \ldots.0 \\ \vdots & & \vdots & & & \vdots & & \\ 0, & 0, & \ldots & 0, & y_0, & y_1, & \ldots.; & y_g \end{vmatrix}$$

so that the j^{th} row of Y is

$$(\underset{j-1 \text{ zeros}}{0, 0, \ldots 0}, y_0, y_1, \ldots, y_g, \underset{n+1-j \text{ zeros}}{0, 0, \ldots 0})$$

Then, from equation 1.09,

$$\underline{R} = \underline{S}\, Y + \underline{W}$$

$\underline{S}$ is a vector with n + 1 components, each of which has m possible values. The number of possible values of $\underline{S}$ is therefore m^{n+1}. Let these values be denoted $\underline{X}_j$ for $j = 1, 2, \ldots, m^{n+1}$. Then it may be shown [6] that the vector $\underline{X}_j$ with greatest probability of being equal to $\underline{S}$, is the one for which

$$|\underline{R} - \underline{X}_j\, Y|$$

is minimised. (Where $|\underline{R} - \underline{X}_j\, Y|$ is the Euclidean distance between the vectors $\underline{R}$ and $\underline{X}_j\, Y$). This vector $\underline{X}_j$ is then said to give the optimum estimate of the data sequence $s_0, s_1, \ldots, s_n$.

1.16 An Advanced Decision Feedback Equalizer

From equation 1.09, the received signal sequence is given by

$$r_i = \sum_{h=0}^{g} y_n \, s_{i-h} + w_i$$

for i = 0, 1, 2, ..., where $(y_0, y_1, \ldots, y_g)$ is the sampled impulse response of the channel. $\{s_i\}$ is, of course, the data sequence and w_i is a sample from a white Gaussian waveform. s_i is defined to be zero for $i < 0$.

First consider the detection of s_0. The received samples $r_0, r_1, \ldots, r_g$ each contain information about s_0 (see equation 1.09). Hence there is no reason why only one of them should be used in the detection of s_0, as in the case of the decision feedback equalizer described in Section 1.12. With the more sophisticated version of this detector, any number p + 1, of samples, may be used in the detection of each data element.

Let

$$\underline{R} = (r_0, r_1, \ldots, r_p),$$

$$\underline{S} = (s_0, s_1, \ldots, s_p),$$

$$\underline{W} = (w_0, w_1, \ldots, w_p)$$

and let Y be the (p + 1) x (p + 1) upper triangular matrix given by

$$Y = \left| \begin{array}{l} y_0, y_1, \ldots\ldots\ldots, y_g, 0, \ldots\ldots\ldots\ldots\ 0 \\ 0\ , y_0, y_1, \ldots\ldots\ldots, y_g, 0\ldots\ldots\ldots\ 0 \\ 0, 0, y_0, y_1, \ldots\ \ldots\ldots\ldots, y_g, 0, \ldots\ 0 \\ \\ \qquad\qquad\qquad\quad y_0, y_1, \ldots\ldots\ldots\ y_g \\ \qquad\qquad\qquad\qquad\qquad\qquad\quad \vdots \\ \qquad\qquad\qquad\qquad\qquad\qquad y_0\ y_1 \\ \qquad\qquad\qquad\qquad\qquad\qquad\quad y_0 \end{array} \right|$$

Then, from equation 1.09,

$$\underline{R} = \underline{S}\ Y + \underline{W} \tag{1.18}$$

$\underline{S}$ is a vector with p + 1 components, each of which has m possible values. Let these values be denoted $\underline{X}_j$, for j = 1, 2, ..., m^{p+1}. Then it may be shown [6] that the vector $\underline{X}_j$ which has the highest probability of being equal to $\underline{S}$ when $\underline{R}$ is given, is the one for which

$$|\underline{R} - \underline{X}_j\ Y|$$

is a minimum. This quantity may be evaluated for each value of j, to give the optimum estimate $\underline{X}_j$ of $\underline{S}$. In this detection process, the first component of the optimum $\underline{X}_j$ is taken as the detected value of s_0, and the other components are discarded.

From equation 1.09,

$$r_i = \sum_{h=0}^{g} y_h \, s_{i-h} + w_i$$

for $i = 0, 1, 2, \ldots\ldots$, where $s_i = 0$ for $i < 0$.

$$\therefore \quad r_1 = y_0 \, s_1 + y_1 \, s_0 + w_1$$

$$r_2 = y_0 \, s_2 + y_1 \, s_1 + y_2 \, s_0 + w_2$$

$$\vdots$$

$$r_g = y_0 \, s_g + y_1 \, s_{g-1} + \ldots + y_g \, s_o + w_g.$$

$$\vdots$$

$$r_{p+1} = y_0 \, s_{p+1} + y_1 \, s_p + \ldots + y_g \, s_{p+1-g} + w_{p+1}$$

These equations can be rewritten in the form

$$\left|\begin{matrix} r_1 \\ r_2 \\ \cdot \\ \\ r_{p+1} \end{matrix}\right|^T = s_0 \left|\begin{matrix} y_1 \\ y_2 \\ \vdots \\ y_g \\ 0 \\ \vdots \\ 0 \end{matrix}\right|^T + (s_1, s_2, \ldots, s_{p+1}) \, Y + \left|\begin{matrix} w_1 \\ w_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ w_{p+1} \end{matrix}\right|^T$$

where T denotes the transpose of a vector. Now redefine the vectors $\underline{R}$, $\underline{S}$ and $\underline{W}$ as follows:

Let:

$$\underline{R} = \begin{vmatrix} r_1 \\ r_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ r_{p+1} \end{vmatrix}^T - s_0 \begin{vmatrix} y_1 \\ y_2 \\ \vdots \\ y_g \\ 0 \\ \vdots \\ 0 \end{vmatrix}^T, \ \underline{S} = \begin{vmatrix} s_1 \\ s_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ s_{p+1} \end{vmatrix}^T \text{and } \underline{W} = \begin{vmatrix} w_1 \\ w_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ w_{p+1} \end{vmatrix}^T$$

Then it can be seen, that the above equation can be written in the form

$$\underline{R} = \underline{S}\ Y + \underline{W}$$

and that equation 1.18 holds for these new vectors $\underline{R}$, $\underline{S}$ and $\underline{W}$.

Now denote the m^{p+1} values of the new vector $\underline{S}$ by $\underline{X}_j$, for $j = 1, 2, \ldots, m^{p+1}$. Then an optimum estimate $\underline{X}_j$, of $\underline{S}$, can be found as before, and the first component of this estimated vector $\underline{S}$, taken as the detected value of s_1. The process continues in this way until the complete data sequence has been detected.

When s_0 is detected, the vector $\underline{R}$ is redefined by

$$\underline{R} = \begin{vmatrix} r_1 \\ r_2 \\ \\ \\ \\ r_{p+1} \end{vmatrix}^T - s_0 \begin{vmatrix} y_1 \\ y_2 \\ \vdots \\ y_g \\ 0 \\ \vdots \\ 0 \end{vmatrix}^T$$

This subtraction of vectors is performed by means of a delay and buffer store, as shown in Figure 1.11. Consider the stage of the detection process in which the data element s_j is about to be detected. The output from the delay at this time is s_{j-1}, which is multiplied by the p+1 component vector

$$(y_1, y_2, \ldots, y_g, 0, \ldots, 0)$$

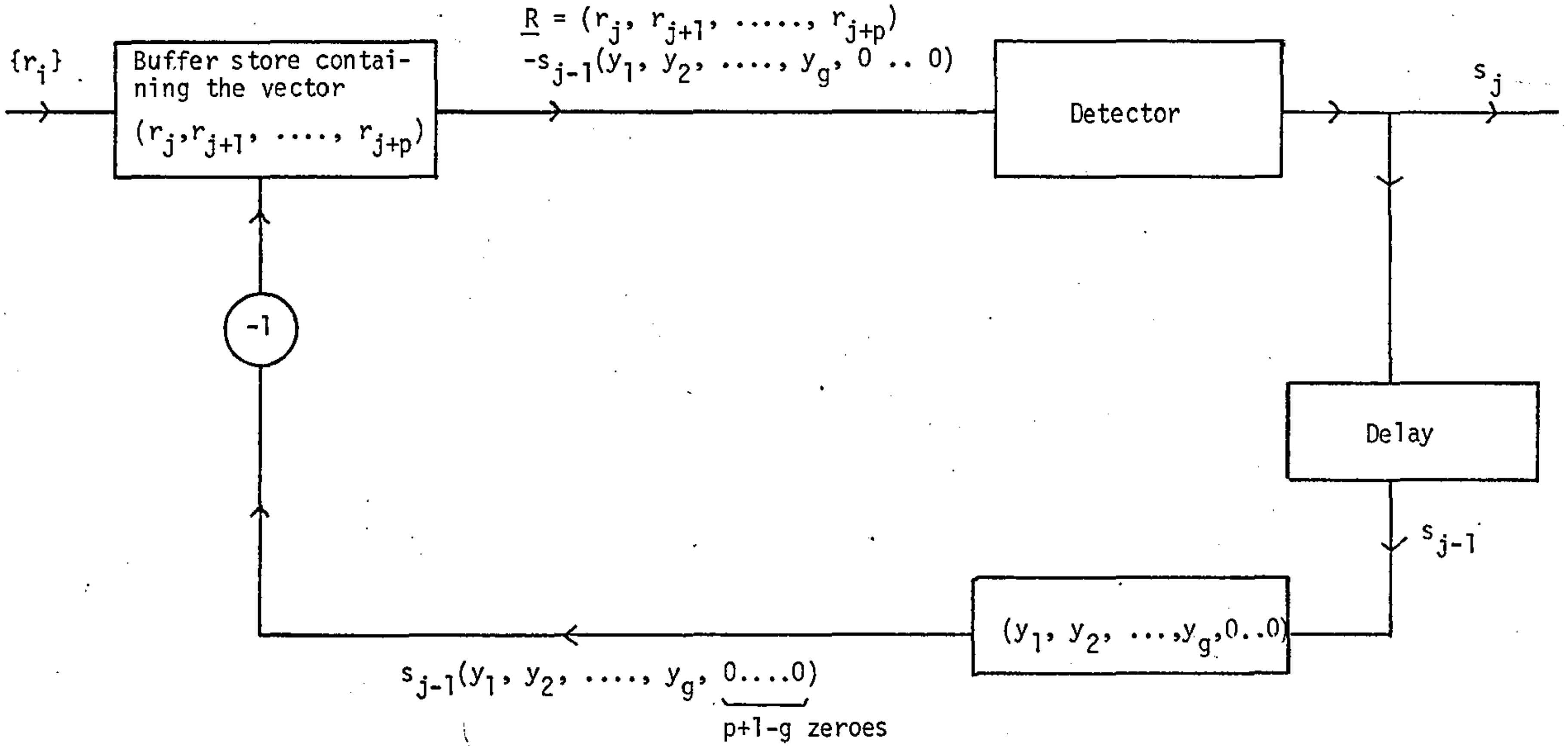
The resulting vector is then subtracted from

$$(r_j, r_{j+1}, \ldots, r_{j+p})$$

which is held in a buffer store, so that the new vector $\underline{R}$ is formed. The detector then selects the vector $\underline{X}_j$ such that

$$|\underline{R} - \underline{X}_j Y|$$

is minimised, and detects s_j from the first component of this vector.



{r_i}
Buffer store containing the vector
(r_j, r_j+1, ..., r_j+p)
R = (r_j, r_j+1, ..., r_j+p)
-s_j-1(y_1, y_2, ..., y_g, 0 .. 0)
Detector
s_j
Delay
s_j-1
(y_1, y_2, ...,y_g,0..0)
-1
s_j-1(y_1, y_2, ..., y_g, 0....0)
p+1-g zeroes

FIGURE 1.11 An advanced decision feedback equalizer

CHAPTER 2

2.01 Data Transmission without Intersymbol Interference

Consider the model of a data transmission system described in Section 1.02, in which the data elements s_i modulate a series of unit impulses $\delta(t)$. Each s_i can take on the m values:

$$-m + 1, \quad - m + 3, \quad \ldots.., m - 1$$

for some given even integer m. The transmitted signal then takes the form

$$\sum_{i=0}^{n} s_i \ \delta(t - iT)$$

where T is the time interval between successive data elements s_i. The transmission channel is a linear baseband channel with impulse response $y(t)$. Let the duration of $y(t)$ be T*, so that $y(t)$ is only non zero for $0 \leq t \leq T^*$.

Now consider a case where there is no noise introduced by the system, and the interval T between successive data elements being transmitted, is greater than T*. Then the response of the channel to one data element will have died away before the next element is transmitted, and there will be no intersymbol interference (i.e. no overlapping of signals). The channel is linear, so an input $s_0 \ \delta(t)$ will give rise to an output $s_0 \ y(t)$. To determine the value of s_0, the detector can compare the output $s_0 \ y(t)$ with the m possible outputs $s\,y(t)$, where s may take the values

$$-m + 1, \quad -m + 3, \quad \ldots.., m - 1.$$

The value of s for which $s\,y(t) = s_0 \ y(t)$ is then taken as the detected value of s_0.

Now consider the case where a white Gaussian noise waveform is added to the signal, at the end of the transmission path, as shown in Figure 1.01. Then, for an input $s_0\ \delta(t)$, the output from the channel is given by

$$r(t) = s_0\ y(t) + w(t) \tag{2.01}$$

where the value of $w(t)$, at any time, is a sample from a normal distribution with zero mean.

Let the maximum value of the impulse response $y(t)$ occur at $t = t_0$, and let $y_0 = y(t_0)$. Also let

$$r_0 = r(t_0) \text{ and}$$

$$w_0 = w(t_0).$$

Then, from equation 2.01,

$$r_0 = s_0\ y_0 + w_0.$$

In general, the sampled response r_i of the received signal at time $t_0 + i\,T$ is given by

$$r_i = s_i\ y_0 + w_i \tag{2.02}$$

where $w_i = w(t_0 + i\,T)$.

Now, if $w_i = 0$, r_i may be compared to $s\ y_0$, for s taking the values $-m + 1$, $- m + 3$, ..., $m - 1$. Then s_i may be detected as the value of s for which

$$r_i = s\, y_0.$$

In general, the noise samples w_i are non zero, and are random variables. It is then not possible to derive the transmitted sequence $\{s_i\}$ with certainty, from the received sequence $\{r_i\}$. It is however possible to find the sequence $\{s_i'\}$, which has the greatest probability of being equal to the transmitted data sequence.

2.02 Maximum Likelihood Detection

Now consider a more general situation than the one described in Section 2.01, in which the transmitted data sequence:

$$s_0, s_1, \ldots\ldots, s_n$$

gives rise to a received sequence:

$$r_0, r_1, \ldots\ldots, r_{n+g}$$

where $g \geq 0$.

Let

$$f(r_0, r_1, \ldots, r_{n+g}/s_0, s_1, \ldots, s_n)$$

be the joint probability density function (pdf) of the random variables

$$r_0, r_1, \ldots, r_{n+g}$$

when

$$s_0, s_1, \ldots, s_n$$

are given.

Assume that the m possible values of each data element s_i are equally likely. Then the maximum likelihood sequence

$$s_o', s_1', \ldots, s_n'$$

when the received sequence $\{r_i\}$ is given, may be defined to be the sequence $\{s_i\}$ which maximises the function

$$f(r_o, r_1, \ldots, r_{n+g}/s_o, s_1, \ldots, s_n)$$

A detector which produces the maximum likelihood sequence is called a maximum likelihood detector.

Theorem 2.01

Assume that the m possible values of the data elements s_i are equally likely and statistically independent. Then the maximum likelihood sequence $\{s_i'\}$, is the estimate of the transmitted data sequence, which has the least probability of being in error.

Proof:

Let

$$\underline{r} = (r_o, r_1, \ldots, r_{n+g})$$

$$\underline{s} = (s_o, s_1, \ldots, s_n)$$

g_1 $(\underline{s}'/\underline{r}')$ = Prob $(\underline{s} = \underline{s}'$ given that $\underline{r} = \underline{r}')$

g_2 $(\underline{r}')$ = Probability density function (pdf) of $\underline{r}$, or the joint pdf of $r_o, r_1, \ldots, r_{n+g}$

$g_3\ (\underline{s}') = \text{Prob}\ (\underline{s} = \underline{s}')$

Then

$$g_1\ (\underline{s}'/\underline{r}')\ g_2\ (\underline{r}') = f\ (\underline{r}'/\underline{s}')\ g_3\ (\underline{s}') \qquad (2.03)$$

(see appendix 1).

The m possible values of the data elements s_i have been assumed to be equally likely and statistically independent, so all sequences

$$s_0,\ s_1,\ \ldots,\ s_n$$

have a probability of $(1/m)^{n+1}$ of occurring. Therefore equation 2.03 gives

$$g_1\ (\underline{s}'/\underline{r}')\ g_2\ (\underline{r}') = f(\underline{r}'/\underline{s}')\ (1/m)^{n+1}.$$

The only terms which depend on $\underline{s}'$ are

$$g_1\ (\underline{s}'/\underline{r}') \qquad \text{and} \qquad f(\underline{r}'/\underline{s}').$$

Hence the maximum likelihood sequence $\underline{s}'$, which maximises $f(\underline{r}'/\underline{s}')$ for given $\underline{r}'$, also maximises $g_1\ (\underline{s}'/\underline{r}')$. But

$$g_1\ (\underline{s}'/\underline{r}') = \text{Prob}\ \ (\underline{s} = \underline{s}' \text{ given that } \underline{r} = \underline{r}'),$$

so the maximum likelihood sequence $\underline{s}'$ also maximises

$$\text{Prob.}\ (\underline{s} = \underline{s}' \text{ given that } \underline{r} = \underline{r}').$$

Therefore the maximum likelihood sequence $\underline{s}'$, is the one which has the greatest probability of being correct.

End of proof.

Now return to the problem of Section 2.01. From equation 2.02,

$$r_i = s_i \, y_0 + w_i$$

where w_i is N $(0, \sigma^2)$ i.e. w_i is a normally distributed random variable with zero mean and some variance σ^2. Therefore r_i is $N(s_i \, y_0, \sigma^2)$ if s_i is given.

A random variable X which is $N(\mu, \sigma^2)$, has a probability density function (pdf) given by

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(- \frac{(x - \mu)^2}{2\sigma^2}\right)$$

Hence the pdf of r_i, when s_i is known, is given by

$$f_1 \, (r_i'/s_i) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(- \frac{(r_i' - s_i \, y_0)^2}{2\,\sigma^2}\right) \qquad (2.04)$$

For this particular problem r_i depends only on s_i, and not on any of the other data elements. (See equation 2.02). Hence the received signal samples r_i are independent of each other, and the conditional pdf of $\underline{r}$, when it is given that $\underline{s} = \underline{s}'$, is given by

$$f(\underline{r}'/\underline{s}') = \prod_{i=0}^{n} f_1 \, (r_i'/\underline{s}') \qquad \text{(this is a standard result).}$$

But each r_i depends only on s_i, and not the other data elements.

$$\therefore \; f_1 \, (r_i'/\underline{s}') = f_1 \, (r_i'/s_i')$$

and

$$f\,(\underline{r}'/\underline{s}') = \prod_{i=0}^{n} f_1\,(r_i'/s_i')$$

Now, using equation 2.04 gives

$$f\,(\underline{r}'/\underline{s}') = \prod_{i=0}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\,[- \frac{(r_i' - s_i'\,y_0)^2}{2\sigma^2}]$$

$$= (\frac{1}{\sigma\sqrt{2\pi}})^{(n+1)} \exp\,[- \sum_{i=0}^{n} \frac{(r_i' - s_i'\,y_0)^2}{2\sigma^2}]$$

$$\ell n\, f\,(\underline{r}'/\underline{s}') = \ell n\, [(\frac{1}{\sigma\sqrt{2\pi}})^{n+1}] - \frac{1}{2\sigma^2} \sum_{i=0}^{n} (r_i' - s_i'\,y_0)^2 \tag{2.05}$$

The maximum likelihood sequence $\{s_i'\}$ is the one which maximises $f\,(\underline{r}'/\underline{s}')$, and $\ell n\, x$ is an increasing function of x. Hence this sequence $\{s_i'\}$ also maximises $\ell n\, f\,(\underline{r}'/\underline{s}')$. Therefore, from equation 2.05 the maximum likelihood sequence $\{s_i'\}$ is the one for which

$$\sum_{i=0}^{n} (r_i' - s_i'\,y_0)^2$$

is minimised. In this particular situation, the maximum likelihood estimate can be found by choosing each s_i' separately, in such a way that $(r_i' - s_i'\,y_0)^2$ is minimised.

Note that the usual definition of the distance between two vectors

$$(x_1, x_2, \ldots, x_n) \quad \text{and} \quad (y_1, y_2, \ldots, y_n) \text{ is}$$

$$(\sum_{i=1}^{n} (x_i - y_i)^2)^{\frac{1}{2}}$$

Hence choosing the sequence $\{s_i\}$ which minimises

$$\sum_{i=0}^{n} (r_i' - s_i' \, y_o)^2$$

is equivalent to finding the vector

$$y_o \, (s_o', s_1', \ldots, s_n')$$

which, is closest to

$$(r_o, r_1', \ldots, r_n')$$

Also note that

$$(s_o \;\; y_o, s_1 \, y_o, \ldots., s_n \, y_o)$$

represents the set of all possible received signal vectors, in the absence of noise. (Let $w_i = 0$ for $i = 0, 1, \ldots, n$ in equation 2.02). Hence the maximum likelihood sequence $\{s_i\}$, may be found by considering the set of all possible received signal vectors, in the absence of noise. Then the vector from this set which is closest to the actual received signal vector, is the one corresponding to the maximum likelihood sequence $\{s_i\}$.

2.03 Data Transmission with Intersymbol Interference

Now consider a case where the time T, of the separation between transmitted digits, is less than the time T* that it takes the channel's impulse response to decay to zero. Then the output of the channel at any time will depend upon several of the transmitted digits, thus giving intersymbol interference (i.e. overlapping of signals).

Let the input to the channel be described by the function

$$\sum_{h=0}^{n} s_h \, \delta(t - hT)$$

as before, so that the data elements s_i are transmitted at intervals of time T. The channel's impulse response is y(t) (see Section 1.02), so the output at time t is given by

$$r(t) = \sum_{h=0}^{n} s_h \, y(t - hT) + w(t) \qquad (2.06)$$

(this is the same as equation 1.02)

Let p and q be the smallest and largest integers respectively, such that

$$y(pT) \neq 0 \quad \text{and}$$

$$y(qT) \neq 0.$$

Let

$$g = q - p$$

and $y_i = y\,[(p + i)T]$ for $i = 0, 1, \ldots, g$.

Then

$$(y_o, y_1, \ldots, y_g)$$

is called the sampled impulse response of the channel, when sampled at intervals of T seconds (see Figure 2.01). T is the interval between successive data elements being transmitted, so the number of components of the sampled impulse response depends on the rate of transmission of data. (The number of components also depends, of course, on the duration of the impulse response y(t)).

Now let

$$r_i = r\ [(p + i)T] \qquad \text{and}$$

$$w_i = w\ [(p + i)T]$$

Then

$$r_i = \sum_{h=i-g}^{i} s_h\, y_{i-h} + w_i \tag{2.07}$$

where s_i has some fixed given value for $i < 0$ or $i > n$ (see equation 1.09),

or

$$r_i = z_i + w_i \tag{2.08}$$

where

$$z_i = \sum_{h=i-g}^{i} s_h\, y_{i-h} \tag{2.09}$$

Now suppose that a random variable X is normally distributed with mean μ and variance σ^2, i.e. X is $N(\mu, \sigma^2)$. Then it is a standard

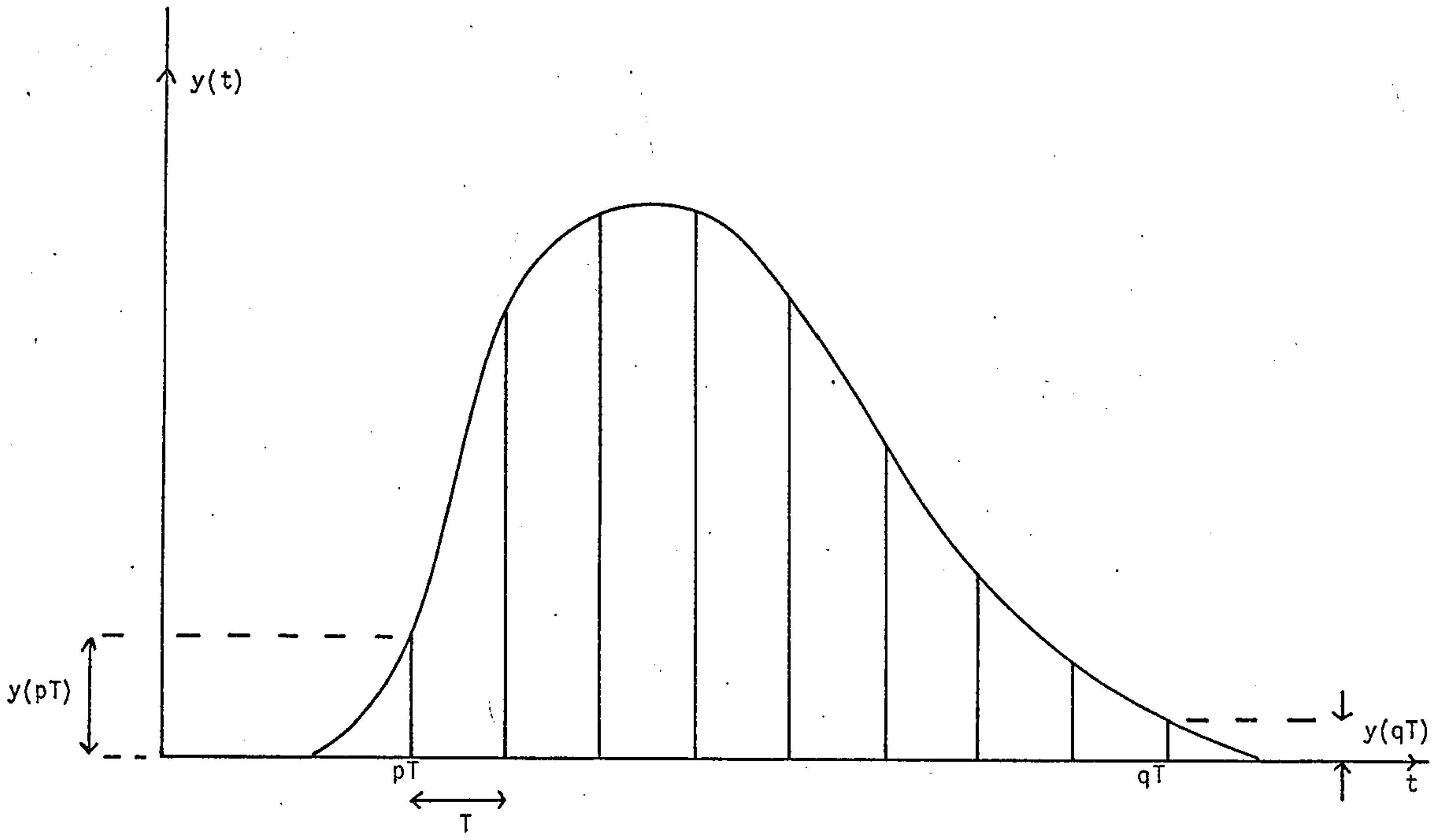


FIGURE 2.01
A channel's impulse response

result that X + c is N(μ + c, σ^2), for any given value c. Let the data sequence $\{s_i\}$ be given, so that the sequence $\{z_i\}$ is also given. w_i is assumed to be N(0, σ^2) for some value of σ, and

$$r_i = z_i + w_i$$

(see equation (2.08)), $\therefore$ r_i is N(z_i, σ^2).

Hence, if $\{s_i\}$ is given, r_i has a probability density function given by

$$f_k\ (r_k'/\{s_i\}) = \frac{1}{\sigma\sqrt{2\pi}}\ \exp[-\frac{(r_k - z_k)^2}{2\ \sigma^2}] \qquad (2.10)$$

and the samples r_i are independent random variables.

It is also a standard result that, for a sequence of independent random variables $\{r_i\}$, the joint probability density function (pdf) of

$$r_0, r_1, \ldots, r_{n+g}$$

satisfies the equation

$$f(r_0', r_1', \ldots, r_{n+g}') = \prod_{i=0}^{n+g} f_i(r_i')$$ where Π denotes the product of the n+g+1 terms

and $f_i(r_i')$ is the pdf of r_i.

Now let

$$f(r_o', r_1', \ldots, r'_{n+g}/s_o, s_1, \ldots, s_n) \text{ or } f(r_o', r_1', \ldots, r'_{n+g}/\{s_i\})$$

be the joint pdf of

$$r_o, r_1, \ldots, r_{n+g}$$

when the sequence $\{s_i\}$ is given. Then this joint pdf satisfies the equation

$$f(r_o', r_1', \ldots, r'_{n+g}/\{s_i\}) = \prod_{j=0}^{n+g} f_j\,(r_j'/\{s_i\}).$$

where $f_j(r_j'/\{s_i\})$ is the pdf of r_j,

when $\{s_i\}$ is given.

z_i is a function of the terms of the sequence $\{s_i\}$ so $\{z_i\}$ is given when $\{s_i\}$ is given.

Therefore

$$f(r_o', r_1', \ldots, r'_{n+g}/\{s_i\}) = \prod_{i=0}^{n+g} f_j(r_j'/\{z_i\})$$

$$= \prod_{i=0}^{n+g} f_j(r_j'/z_j)$$

as r_j depends only on z_j and not on the other terms of the series $\{z_i\}$, (see equation 2.08). Hence from equation 2.10,

$$f(r_0', r_1', \ldots, r'_{n+g}/\{s_i\}) = \prod_{i=0}^{n+g} [\frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(r_k' - z_k)^2}{2\sigma^2})]$$

or

$$f(\underline{r}'/\underline{s}) = \prod_{i=0}^{n+g} [\frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{(r_k' - z_k)^2}{2\sigma^2})]$$

where

$$\underline{r}' = (r_0', r_1', \ldots, r'_{n+g})$$

and

$$\underline{s} = (s_0, s_1, \ldots, s_n).$$

$$\therefore \quad \ell n\, f(\underline{r}'/\underline{s}) = \ell n\, [(\frac{1}{\sigma\sqrt{2\pi}})^{n+g+1}] - \sum_{i=0}^{n+g} \frac{(r_{k'} - z_k)^2}{2\sigma^2} \qquad (2.11)$$

$\ell n(x)$ is an increasing function of x, so $\ell n\, f(\underline{r}'/\underline{s})$ is maximised when $f(\underline{r}'/\underline{s})$ is maximised. Hence the maximum likelihood sequence $\{s_i'\}$, is the sequence which minimises

$$\sum_{k=0}^{n+g} (r_k - z_k)^2$$

when $\{r_i\}$ is given, where

$$z_i = \sum_{h=i-g}^{i} s_h\, y_{i-h}.$$

(this follows from equation (2.11) and the definition of the maximum likelihood sequence, given in Section 2.02).

Now let

$$g(\underline{s}) = \sum_{k=0}^{n+g} (r_k - \sum_{h=k-g}^{k} s_h\, y_{k-h})^2 = \sum_{k=0}^{n+g} (r_k - z_k)^2.$$

Then $g(\underline{s})$ is called the cost function for the sequence

$$s_o, s_1, \ldots, s_n$$

and the problem of finding the maximum likelihood sequence is equivalent to finding the sequence $\{s_i\}$ which minimises $g(\underline{s})$.

For the case of an m level signal, each s_i make take on m possible values. Then

$$g(\underline{s}) \text{ or } g(s_o, s_1, \ldots, s_n)$$

can take on m^{n+1} possible values corresponding to the different possible combinations of the sequence

$$s_o, s_1, \ldots, s_n$$

One method for finding the maximum likelihood sequence is to evaluate $g(\underline{s})$ for each possible data sequence, and select the sequence corresponding to the minimum value of $g(\underline{s})$. It will be shown in the following section, that the problem of finding the maximum likelihood detected sequence, is equivalent to that of finding the shortest path through a given trellis diagram. This shortest path problem can then be solved by a technique known as dynamic programming. A dynamic programming algorithm (the Viterbi Algorithm) will then be described, which can produce the maximum likelihood sequence with much less computation, than that which would be required to evaluate $g(\underline{s})$ for all values of $\underline{s}$.

2.04 The Trellis Diagram

Define $\underline{Q}_k$ by

$$\underline{Q}_k = (s_{k-g}, s_{k-g+1}, \ldots, s_{k-1})$$

for $k = 0, 1, 2, \ldots, n+g+1$, where s_k is defined to be equal to $-m+1$ for $k < 0$ or $k > n$. Clearly, for any sequence

$$s_o, s_1, \ldots., s_n$$

there is only one corresponding sequence

$$\underline{Q}_1, \underline{Q}_2, \ldots, \underline{Q}_{n+g}.$$

Let

$$h(\underline{Q}_1, \underline{Q}_2, \ldots, \underline{Q}_{n+g}) = g(s_o, s_1, \ldots, s_n).$$

Then the problem of minimising $g(\underline{s})$ is transformed to one of finding the sequence

$$\underline{Q}_1, \underline{Q}_2, \ldots, \underline{Q}_{n+g}$$

which minimises

$$h(\underline{Q}_1, \underline{Q}_2, \ldots, \underline{Q}_{n+g}).$$

Each data element s_i can then take on m different values and $\underline{Q}_k$ depends on g of these elements, therefore, $\underline{Q}_k$ can take on m^g different possible values. Now consider the trellis diagram shown in Figure 2.02. In this diagram, each vertical line represents a vector $\underline{Q}_k$, and each horizontal line represents one of the m^g possible vector values that $\underline{Q}_k$ can take on. The intersection of a vertical and a horizontal line is called a node. Now any sequence of vectors can be represented by a path through the trellis, passing through one node for each vector $\underline{Q}_k$.

From the definition of $\underline{Q}_k$, and the fact that $s_i = -m+1$ for $i < 0$ and $i > n$, it can be seen that

$$\underline{Q}_0 = (-m+1, -m+1, \ldots, -m+1)$$

and

$$\underline{Q}_{n+g+1} = (-m+1, -m+1, \ldots, -m+1).$$

Hence the path through the trellis must start at point A, on the diagram, and finish at point B. (Both of these points lie on the horizontal line representing the appropriate vector value).

Example 2.01

Now consider a channel with sampled impulse response

$$(a_0, a_1, a_2)$$

being used in conjunction with a binary data signal, so that each s_i has the possible values ± 1. For this channel, g = 2 so

$$\underline{Q}_k = (s_{k-2}, s_{k-1}).$$

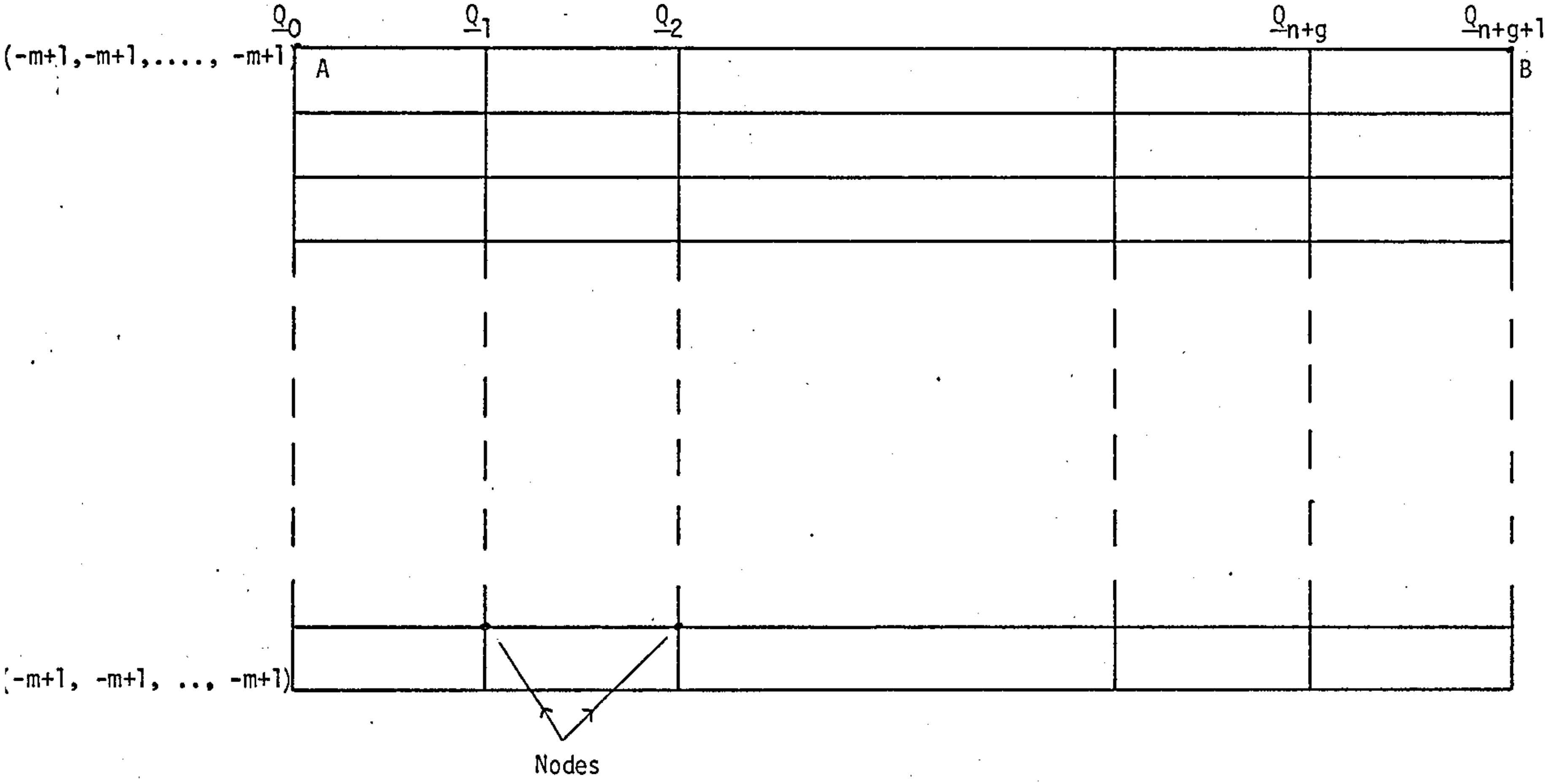


FIGURE 2.02

A trellis diagram

Suppose that $n = 4$ so that the transmitted message is

$$(s_o, s_1, \ldots, s_4)$$

with s_i defined to be equal to -1

for $i < 0$ or $i > 4$. Then

$$\underline{Q}_o = (-1, -1) \quad \text{and}$$

$$\underline{Q}_7 = (-1, -1)$$

$\underline{Q}_k$ can take the values

$$(1, 1), \quad (1, -1), \quad (-1, 1) \quad \text{and} \quad (-1, -1)$$

for $k = 2, 3, 4$ and 5

Let

$$(s_o, s_1, s_2, s_3, s_4) = (-1, 1, -1, -1, 1)$$

for this example. Then

$$\underline{Q}_1 = (-m+1, s_o) = (-1, -1)$$

$$\underline{Q}_2 = (s_o, s_1) = (-1, 1)$$

$$\underline{Q}_3 = (s_1, s_2) = (1, -1)$$

$$\underline{Q}_4 = (s_2, s_3) = (-1, -1)$$

$$\underline{Q}_5 = (s_3, s_4) = (-1, 1) \quad \text{and}$$

$$\underline{Q}_6 = (s_4, -m+1) = (1, -1).$$

Hence the path through the trellis diagram shown in Figure 2.03, exactly represents the transmitted data sequence $\{s_i\}$. Note that, for any given vector

$$\underline{Q}_i = (s_{i-2}, s_{i-1})$$

there are only two possible following vectors

$$\underline{Q}_{i+1} = (s_{i-1}, s_i),$$

as s_{i-1} is given, and s_i can take on only two values in this example. Figure 2.04 shows the transitions that are possible from any vector $\underline{Q}_i$ to a succeeding vector $\underline{Q}_{i+1}$.

Now return to the general case of an m level signal, and a channel whose sampled impulse response has g+1 components, so that

$$\underline{Q}_k = (s_{k-g}, s_{k-g+1}, \ldots, s_{k-1}).$$

Then, if $\underline{Q}_k$ is given, $s_{k-g+1}, s_{k-g+2}, \ldots, s_{k-1}$ will be given, and the following vector:

$$\underline{Q}_{k+1} = (s_{k-g+1}, s_{k-g+2}, \ldots, s_{k-1}, s_k)$$

can take on m values corresponding to the m possible values of s_k. Hence, from any node on the general trellis (Figure 2.02) representing a vector $\underline{Q}_k$, there are only m possible following nodes representing $\underline{Q}_{k+1}$.

It was shown in Section 2.03, that the maximum likelihood detected sequence $\{s_i\}$, is the sequence which minimises the function

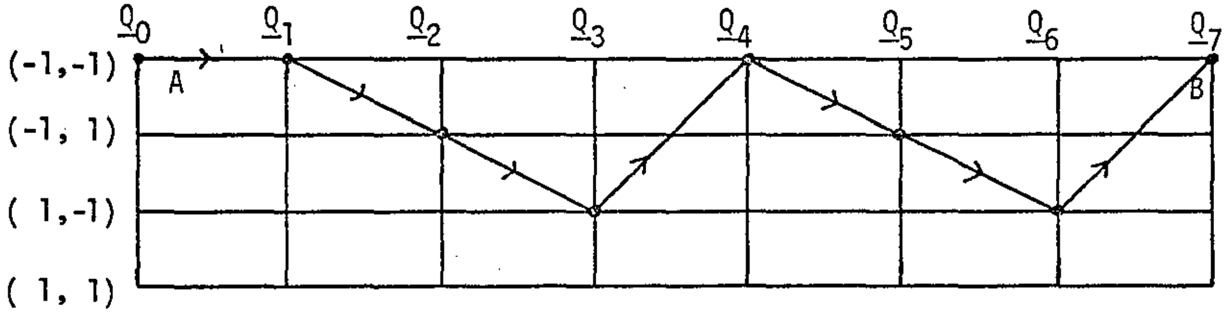


FIGURE 2.03
Trellis for example 2.01

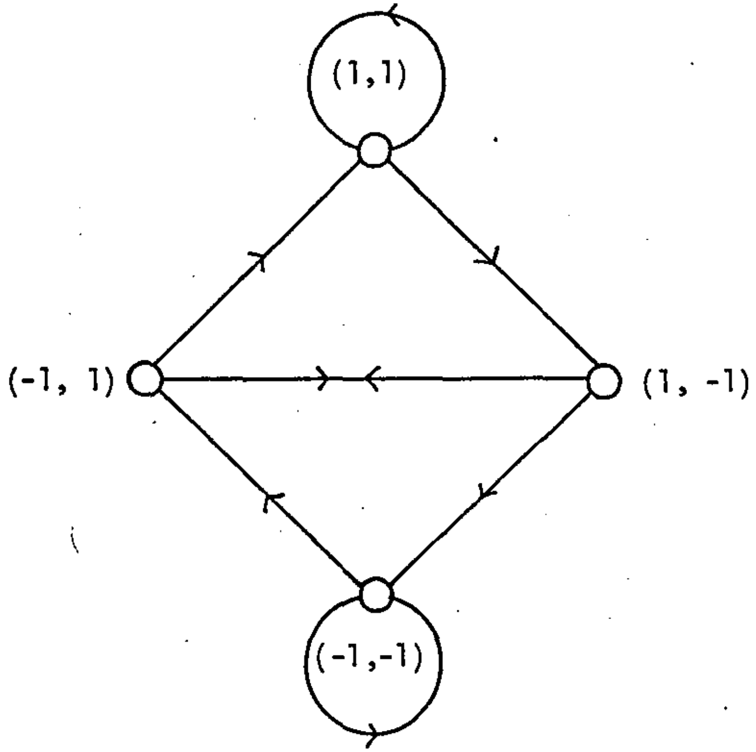


FIGURE 2.04
Transitions from one vector to another

$$\sum_{k=0}^{n+g} (r_k - \sum_{h=k-g}^{k} s_h y_{k-h})^2$$

where s_i has some given fixed value for $i < 0$ or $i > n$. This function may also be written as

$$\sum_{k=0}^{n+g} d_k^{(1)} (s_{k-g}, s_{k-g+1}, \dots, s_k)$$

where

$$d_k^{(1)} (s_{k-g}, s_{k-g+1}, \dots, s_k) = (r_k - \sum_{h=k-g}^{k} s_h y_{k-h})^2$$

Now note that there is only one pair of vectors $(\underline{Q}_k, \underline{Q}_{k+1})$ corresponding to the data elements

$$s_{k-g}, s_{k-g+1}, \dots, s_k$$

Hence

$$(r_k - \sum_{h=k-g}^{k} s_h y_{k-h})^2$$

is a function of $\underline{Q}_k$ and $\underline{Q}_{k+1}$, and the maximum likelihood sequence may be found by minimising the function

$$\sum_{k=0}^{n+g} d_k^{(2)} (\underline{Q}_k, \underline{Q}_{k+1})$$

where

$$d_k^{(2)} (\underline{Q}_k, \underline{Q}_{k+1}) = (r_k - \sum_{h=k-g}^{k} s_h y_{k-h})^2.$$

(It is assumed here that the sequence $\{r_i\}$ is known).

Let the distance between two nodes on the trellis diagram, representing $\underline{Q}_k$ and $\underline{Q}_{k+1}$, be defined to be equal to $\underline{d}_k^{(2)}$ $(\underline{Q}_k, \underline{Q}_{k+1})$. Then any path through the trellis, through the sequence of nodes representing

$$\underline{Q}_o, \underline{Q}_1, \ldots., \underline{Q}_{n+g+1}$$

has length

$$d_o^{(2)} \ (\underline{Q}_o, \underline{Q}_1) + d_1^{(2)} \ (\underline{Q}_1, \underline{Q}_2) + \ldots + d_{n+g}^{(2)} \ (\underline{Q}_{n+g}, \underline{Q}_{n+g+1})$$

or

$$\sum_{i=0}^{n+g} d_k^{(2)} \ (\underline{Q}_k, \underline{Q}_{k+1}).$$

But it has been shown above that the maximum likelihood sequence $\{s_i\}$ may be found, by finding the sequence of vectors $\{\underline{Q}_i\}$ which minimises

$$\sum_{k=0}^{n+g} d_k^{(2)} \ (\underline{Q}_k, \underline{Q}_{k+1}).$$

Hence this maximum likelihood sequence may be found, by finding the sequence of vectors $\{\underline{Q}_i\}$, which minimises the path length through the trellis diagram from point A to point B. i.e. the problem of maximum likelihood detection has been reduced to a shortest path problem.

2.05 The Viterbi Algorithm Applied to a Particular Example

Now consider again, the problem of example 2.01, in which a two level signal is used with a channel whose sampled impulse response is (a_0, a_1, a_2). Let u_k (I, J) be the length of the shortest path from point A (on Figure 2.03) to the node $(I, J)_k$. This is the node representing the possible vector value (I, J) of $\underline{Q}_k$. Also let the sequence of vectors $\{\underline{Q}_i\}$, which gives the shortest path to the node $(I, J)_k$ be denoted

$$\{\underline{P}_i\ (k, I, J)\}$$

Note that each of the vectors in this sequence is a function of k, I and J.

The Viterbi Algorithm (V.A) is a dynamic programming algorithm, which sets up a relationship between the shortest paths to the four $\underline{Q}_k$ nodes, and the shortest paths to the four $\underline{Q}_{k+1}$ nodes. This is repeated for all values of k until point B, on the trellis, is reached.

For this example,

$$\underline{Q}_k = (s_{k-2}, s_{k-1})$$

and $s_i = -1$ for $i < 0$ or $i > 4$. Therefore

$$\underline{Q}_o = (-1, -1).$$

From Figure 2.04, it can be seen that there are only two possible values for $\underline{Q}_1$.

Q_1 can be either

$(-1, -1)$ or $(-1, 1)$.

The sequence of vectors $\{\underline{Q}_i\}$, giving the shortest path from point A to the node

$$(-1, -1)_1$$

is $\{(-1, -1), (-1, -1)\}$

(This is, in fact, the only path from point A to this node).

i.e.

$$\{\underline{P}_i\ (1, -1, -1)\} = \{(-1, -1), (-1, -1)\} \qquad (2.12)$$

Similarly

$$\{\underline{P}_i\ (1, -1, 1)\} = \{(-1, -1), (-1, 1)\} \qquad (2.13)$$

The length of the path to the node

$$(-1, -1)_1$$

is the distance between the nodes

$$(-1, -1)_0 \quad \text{and} \quad (-1, -1)_1$$

which is denoted

$$d_0^{(2)}\ [(-1, -1), (-1, -1)]$$

(see Section 2.04), i.e.

$$u_1\ (-1, -1) = d_0^{(2)}\ [(-1, -1), (-1, -1)] \qquad (2.14)$$

Similarly the length of the shortest path to the node

$$(-1, 1)_1$$

is given by

$$u_1\ (-1,\ 1) = d_0^{(2)}\ [(-1,\ -1),\ (-1,\ 1)] \qquad (2.15)$$

The shortest paths to the four $\underline{Q}_2$ nodes will now be derived. From Figure 2.04, the node

$$(-1,\ -1)_2$$

can only be reached from the two nodes

$$(-1,\ -1)_1 \quad \text{and} \quad (1,\ -1)_1$$

However the node

$$(1,\ -1)_1$$

is not allowable as it cannot be reached from point A of the trellis. Hence the sequence of vectors giving the shortest (and only) path from point A, to the node

$$(-1,\ -1)_2$$

is given by

$$\{\underline{P}_i\ (2,\ -1,\ -1)\} = \{(-1,\ -1),\ (-1,\ -1),\ (-1,\ -1)\} \qquad (2.16)$$

The length of the shortest path to the node

$$(-1,\ -1)_2$$

is equal to the length of the shortest path to the preceeding node $(-1, -1)_1$

+

the distance between the two nodes.

The length of the shortest path to the node

$(-1, -1)_1$ is given in equation (2.14).

Hence

$$u_2\ (-1, -1) = u_1\ (-1, -1) + d_1^{(2)}\ [(-1, -1), (-1, -1)] \quad (2.17)$$

The shortest path to the node

$(-1, -1)_2$,

and the length of this path, have now been found and are given by equations 2.16 and 2.17 respectively. The shortest paths to the other $\underline{Q}_2$ nodes can be found in a similar manner.

For $k > 2$, each $\underline{Q}_k$ node can be reached from two $\underline{Q}_{k-1}$ nodes. The node

$(I, J)_k$

can be reached from the nodes

$(-1, I)_{k-1}$ and $(1, I)_{k-1}$

(see Figure 2.04). Hence the shortest path to the node

$(I, J)_k$ is either:

(a) the shortest path to the node

$$(-1, I)_{k-1}$$

+ the distance between the nodes

$$(-1, I)_{k-1} \quad \text{and} \quad (I, J)_k$$

or

(b) the shortest path to the node

$$(1, I)_{k-1}$$

+ the distance between the nodes

$$(1, I)_{k-1} \quad \text{and} \quad (I, J)_k$$

i.e.

$$u_k (I, J) = \min_{k= \pm 1} [u_{k-1} (K, I) + d_{k-1}^{(2)} [(K, I), (I, J)]] \qquad (2.18)$$

Equation 2.18 holds for I = ±1 and J = ±1. Let the value of K selected here, be K_1. (K_1 will, of course, be different for different values of I and J). Then the sequence $\{\underline{Q}_i\}$, giving the shortest path to the node

$$(I, J)_k$$

is the sequence $\{\underline{Q}_i\}$ giving the shortest path to the node

$$(K_1, I)_{k-1}$$

with the additional vector (I, J)

i.e.

$$\{\underline{P}_i\ (k,\ I,\ J)\} = \{\underline{P}_i\ (k-1,\ K_1,\ I)\},\ (I,\ J) \qquad (2.19)$$

Equations 2.18 and 2.19 form the Viterbi Algorithm (V.A) for the problem of example 1, and they can be used recursively to find the shortest path, through the trellis, from point A to point B.

2.06 Method for Dealing with Long Transmitted Data Sequences

In Section 2.05 the V.A. was presented in a form which was quite suitable for the problem of example 2.01, in which the transmitted data sequence had only five elements. With the V.A. used in this way, no data elements are detected until the shortest path from point A to point B of the trellis is found. This approach is however impracticable, if the data sequence, and hence the trellis diagram, is very long. Large amounts of storage would then be required, to hold the sequence of vectors $\{\underline{Q}_i\}$, giving the shortest paths to the various nodes of the trellis.

One way of overcoming this problem is to store no more than a given number N, of the vectors in the sequences $\{\underline{Q}_i\}$, giving the shortest paths to each node of the trellis. Hence the sequence of vectors representing the shortest path to each $\underline{Q}_k$ node, would consist only of the appropriate values for the vectors

$$\{\underline{Q}_{k-N+1},\ \underline{Q}_{k-N+2},\ \ldots,\ \underline{Q}_k\}.$$

In general, the trellis has m^g nodes (see Section 2.04), so there will be m^g sequences of N vectors $\underline{Q}_i$, to store.

From equation 2.19 it can be seen that the shortest path to some given $\underline{Q}_k$ node, is formed by adding a new vector to the sequence $\{\underline{Q}_i\}$ of vectors, which represents the shortest path to one of the $\underline{Q}_{k-1}$ nodes. The sequence for the $\underline{Q}_{k-1}$ node will contain N vectors representing

$$\{\underline{Q}_{k-N}, \underline{Q}_{k-N+1}, \ldots, \underline{Q}_{k-1}\}.$$

Hence the vector representing $\underline{Q}_{k-N}$ must be deleted in order to store the sequence of N vectors for the $\underline{Q}_k$ node. Before the vector $\underline{Q}_{k-N}$ is deleted, its earliest component (which represents s_{k-N-g}) could be taken as the detected value of the data element s_{k-N+g}. There will however be m^g $\underline{Q}_{k-N}$ nodes, and therefore many vectors representing $\underline{Q}_{k-N}$, which must be deleted from the appropriate sequences of vectors. Each of the vectors representing $\underline{Q}_{k-N}$ has an element representing s_{k-N-g}, which could be used as the detected value for the data element s_{k-N-g}.

Now consider the $\underline{Q}_k$ node which has a shorter possible path to it, from the point A of the trellis, than any of the other $\underline{Q}_k$ nodes. Let $\underline{Q}'_{k-1}$ be the $\underline{Q}_{k-1}$ node which lies on the optimum path from point A to this $\underline{Q}_k$ node. Then a reasonable strategy for detecting s_{k-N-g}, seems to be that of using the earliest component of the vector $\underline{Q}'_{k-N}$, which lies on the shortest path from A to $\underline{Q}'_{k-1}$.

It has been found from simulation tests that, for fairly large values of N, the shortest paths to each of the $\underline{Q}_{k-1}$ nodes, tend to pass through the same $\underline{Q}_{k-N}$ node. In this case there should be no errors in detection, due to the fact that only N nodes, of each path through the trellis, are stored.

2.07 A General Version of the Viterbi Algorithm with Decisions made after a Fixed Delay

Consider again the problem of an m level signal being used with a channel with sampled impulse response

$$(y_0, y_1, \ldots, y_g).$$

This version of the V.A. stores a number of N component vectors denoted

$$\underline{Q}_{j(1)}, \underline{Q}_{j(2)}, \underline{Q}_{j(3)}, \ldots.$$

Each vector $\underline{Q}_j$ (I) has the form

$$(x_{j-N+1}, x_{j-N+2}, \ldots, x_j).$$

where $N \geq g$ and each x_i represents one of the m possible values of the data element s_i. The algorithm begins each cycle of its detection process with m^g such vectors in store, and with one vector corresponding to each possible combination of the g elements

$$x_{j-g+1}, x_{j-g+2}, \ldots, x_j.$$

(This is true except for the first few cycles, when the algorithm is starting up). As before, the transmitted data sequence is denoted

$$\{s_0, s_1, \ldots, s_n\}$$

and s_i is defined to be equal to $-m+1$ for $i < 0$ or $i > n$

Initially the process starts with one N component vector $\underline{Q}_{-1}(1)$ which is given by

$$\underline{Q}_{-1}(1) = (-m+1, -m+1, \ldots, -m+1)$$

This vector is then extended to m vectors, each with N+1 components, by the addition of the component x_o which can take on m values. These m vectors are given by

$$\underline{T_o}(1, x_o) = (\underline{Q}_{-1}(1), x_o).$$

A quantity called the cost function is now defined for the vector $\underline{T_o}(1, x_o)$, by the equation

$$V_o(1, x_o) = [y_o\, x_o + y_1\,(-m+1) + y_2(-m+1) + \ldots + y_g(-m+1) - r_o]^2$$

where r_o is the first received signal sample. This equation can be written as

$$V_o(1, x_o) = (\underline{Y}. [\underline{T_o}(1, x_o)]_{g+1} - r_o)^2$$

where

$$\underline{Y} = (y_g, y_{g-1}, \ldots, y_o)$$

and

$[\underline{T_o}(1, x_o)]_{g+1}$ is the vector formed from the latest g+1 components (i.e. the g+1 components furthest to the right) of $\underline{T_o}(1, x_o)$.

$\underline{Y}.[\underline{T_0}(1, x_0)]_{g+1}$ is, of course, the scalar product of the two vectors. Note that the vector $\underline{Y}$ is the reverse of the channel vector (i.e. the reverse of the vector representing the channel's sampled impulse response).

The next step of the algorithm is to find the vector $\underline{T_0}(1, x_0)$ which has the smallest cost. This is equivalent to finding the value x_0 which gives the lowest value of $v_0(1, x_0)$. The earliest element of this selected vector is then taken as a detected data element. (The earliest component is, of course, the one furthest to the left). This first detected element will have the value -m+1, as $\underline{Q}_{-1}$ (1) was defined to be the N component vector with all components taking the value -m+1.

To complete the first cycle of the algorithm, the latest N elements of the m vectors $\underline{T_0}(1, x_0)$, are stored in the array $\underline{Q}_0(I)$. The m costs $v_0(1, x_0)$ are stored in the array $u_0(I)$, where I = 1, 2, ..., m.

For the second cycle, the m vectors $\underline{Q}_0(I)$ are extended to the m^2 vectors defined by

$$\underline{T_1}(I, x_1) = (\underline{Q}_0(I), x_1)$$

where x_1 can take on the m data element values. The costs for each of these m^2 vectors are given by

$$v_1(I, x_1) = u_0(I) + \{\underline{Y}. [\underline{T_1}(I, x_1)]_{g+1} - r_1\}^2$$

where r_1 is the second received signal sample. As before, the earliest element of the vector $\underline{T_1}(I, x_1)$ with smallest cost, is taken as a detected data element.

The latest N elements of the vectors $\underline{T}_1(I,x_1)$ are then stored in the array $\underline{Q}_1(J)$, where $J = 1, 2, \ldots, m^2$. The corresponding costs $v_1(I, x_1)$, are stored in the array $u_1(J)$, to complete the second cycle of the V.A.

Hence, after two elements have been detected (after two cycles of the algorithm), the number of vectors stored is m^2, where m is the number of signal levels.

The process continues in this way, with.each cycle beginning by extending each of the vectors $\underline{Q}_j(I)$ to the m vectors defined by

$$\underline{T}_{j+1}(I, x_{j+1}) = (\underline{Q}_j(I), x_{j+1}) \qquad (2.20)$$

The costs for these extended vectors are given by

$$v_{j+1}\ (I, x_{j+1}) = u_j(I) + \{\underline{Y}.[\underline{Q}_j(I), x_{j+1}]_{g+1} - r_{j+1}\}^2 \qquad (2.21)$$

where r_{j+1} is the j+2 nd received signal sample. The detected element is, as before, the earliest component of the vector $\underline{T}_{j+1}(I, x_{j+1})$ with smallest cost.

After the (g+1)st element has been detected, there will be m^{g+1} vectors denoted $\underline{T}_g\ (I, x_g)$. Then in this cycle (and every following cycle) of the algorithm, all but m^g of these vectors are deleted from storage, before defining the vectors $\underline{Q}_g(J)$. The m^g vectors to be retained for the next cycle, are selected by keeping the vector with lowest cost, for each possible combination of the latest g components of the vectors. Then the latest N components of the retained vectors $\underline{T}_{j+1}\ (I, x_{j+1})$ are stored in the array $\underline{Q}_{j+1}(J)$, and the costs v_{j+1} (I, x_{j+1}) for the retained vectors, are stored in the array u_{j+1}(J). (J may take the values 1, 2, ..., m^g). Hence each following cycle of the algorithm will begin with m^g vectors stored in an array denoted $\underline{Q}_i$(J), for some integer i. A flowchart for this form of the V.A. is given in Figure 2.05.

Note that, instead of starting the detection process with one vector $\underline{Q}_{-1}$ (1), a full set of m^g vectors, representing each possible combination of the latest g elements, could be used. These vectors would be denoted

$$\underline{Q}_{-1}\,(1),\ \underline{Q}_{-1}\,(2),\ \ldots\ldots,\ \underline{Q}_{-1}\,(m^g).$$

In this case the vector $\underline{Q}_{-1}$ (I), whose latest g components have the value -m+1, should be assigned a cost equal to zero and the other vectors assigned some very large cost. Then all future vectors stemming from the ones with large costs, will also have large costs, and will eventually be deleted from the system. After g cycles of the process, the m^g vectors retained by the algorithm will all have stemmed from the vector $\underline{Q}_{-1}$ (I) with zero cost. These m^g vectors will then be identical to those for the situation where the algorithm is started with just one vector $\underline{Q}_{-1}$ (1).

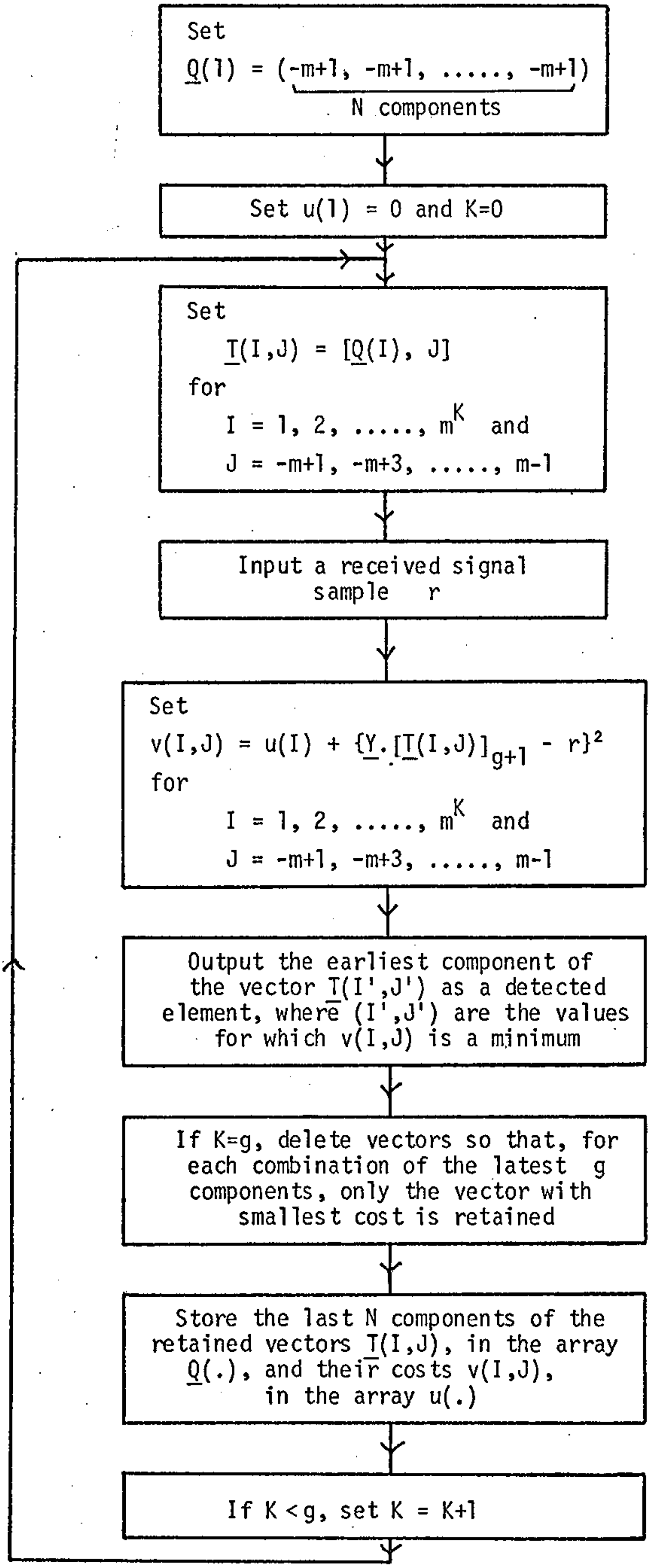

Set
$\underline{Q}(1) = (\underbrace{-m+1, -m+1, \ldots\ldots, -m+1}_{N \text{ components}})$
Set u(1) = 0 and K=0
Set
$\underline{T}(I,J) = [\underline{Q}(I), J]$
for
$I = 1, 2, \ldots\ldots, m^K$ and
$J = -m+1, -m+3, \ldots\ldots, m-1$
Input a received signal sample r
Set
$v(I,J) = u(I) + \{\underline{Y}.[\underline{T}(I,J)]_{g+1} - r\}^2$
for
$I = 1, 2, \ldots\ldots, m^K$ and
$J = -m+1, -m+3, \ldots\ldots, m-1$
Output the earliest component of the vector $\underline{T}(I',J')$ as a detected element, where (I',J') are the values for which v(I,J) is a minimum
If K=g, delete vectors so that, for each combination of the latest g components, only the vector with smallest cost is retained
Store the last N components of the retained vectors $\underline{T}(I,J)$, in the array $\underline{Q}(.)$, and their costs v(I,J), in the array u(.)
If K < g, set K = K+1

FIGURE 2.05
A flow diagram for the Viterbi algorithm

2.08 Starting up Procedure for the V.A.

The data elements s_i are defined to be equal to -m+1 for i < 0. This means that the transmitter must send out a sequence of elements with the value -m+1, before the data sequence

$$s_0, s_1, \ldots, s_n$$

is transmitted. If the number of components of the vectors $\underline{Q}_j(I)$ is N, then N elements with the value -m+1 must be transmitted, and $\underline{Q}_{-1}$ (1) is defined to be the N component vector

$$(-m+1, -m+1, \ldots., -m+1)$$

This sequence of elements, which preceeds the data sequence, is called a training signal. It has been found that, if no training signal is used or the wrong training signal is used, the performance of the detection process will be unaffected, apart from an initial burst of errors [33].

2.09 Number of Operations Required by the Viterbi Algorithm

Apart from the first few cycles, the algorithm has m^g vectors $\underline{Q}_j(I)$ in store, at the start of each cycle. (m is the number of signal levels, and g+1 is the number of components of the channel's sampled impulse response). When a new received signal sample r_{j+1} arrives, these m^g vectors are extended to the m^{g+1} vectors denoted $\underline{T}_{j+1}$ (I, x_{j+1}). The costs for these extended vectors are computed using equation 2.21.

Note that, for a situation where the channel characteristics are constant, all possible values of the terms $y_i\ x_j$ should be stored before the detection process begins. (x_j represents the possible values of the data element s_j). Then no further multiplications are needed to form the scalar product

$$\underline{Y}.[\underline{Q}_j\ (I),\ x_{j+1}]_{g+1}$$

in equation 2.21. One multiplication (or squaring operation) is carried out for the evaluation of each of the m^{g+1} costs $v_{j+1}\ (I,\ x_{j+1})$. Then, for each combination of the latest g elements of the vectors, all but the one vector with lowest cost is deleted from storage. There are m vectors $\underline{T}_{j+1}\ (I,\ x_{j+1})$ containing each combination of the latest g elements, so to find the one with smallest cost requires m-1 comparisons. There are m^g possible combinations of the latest g elements, so (m-1) m^g comparisons must be made by the algorithm, during each cycle. Hence the V.A. must perform m^{g+1} multiplications and (m-1) m^g comparisons, for each data element detected,(apart from the first few). Clearly, if m and g are large (typically m = 4 or 16 and g = 8), a vast number of operations must be performed, per detected element.

Other operations, such as additions and the moving of numbers from one store to another, are also needed during the execution of the algorithm. These have not been considered here, but it is hoped that the number of multiplications and comparisons required, will give a good guide to the complexity of the algorithm.

2.10 Probability of Error for the V.A. Detector when used with the Ideal Channel

From the description of the V.A. given in section 2.07, m^g vectors are stored at the start of each cycle of the algorithm (where m is the number of signal levels and g+1 is the number of components of the channel's sampled impulse response). The channel whose sampled impulse response has just one non zero component, with the value unity, is called the ideal channel. It is ideal in the sense that it causes no change in the transmitted data sequence. For the ideal channel, g=0, so the V.A. has just one vector in store at the start of each of the algorithm's cycles.

The first cycle of the algorithm begins with one N component vector given by

$$\underline{Q}_{-1}(1) = (-m+1, -m+1, \ldots., -m+1)$$

This vector is then extended to the m N+1 component vectors:

$$\underline{T}_0(1, x_0) = (-m+1, -m+1, \ldots, -m+1, x_0)$$

The cost for this vector is given by

$$v_0(1, x_0) = \{\underline{Y}.[\underline{T}_0(1, x_0)]_{g+1} - r_0\}^2$$

for $x_0 = -m+1, -m+3, \ldots., m-1,$

where $\underline{Y}$ is the vector whose components are the reverse of the channel's sampled impulse response, and r_0 is the first received signal sample (see section 2.07).

For the case of the ideal channel, g=0 and $\underline{Y}$ is the scalar with value unity.

$$\therefore \quad v_0(1, x_0) = (x_0 - r_0)^2 \tag{2.22}$$

Only the one of the vectors $\underline{T}_0(1, x_0)$, with smallest cost is retained for the next cycle of the algorithm. Hence only one value of x_0 will be available for the detection of s_0 at a later stage. The retained vector $\underline{T}_0(1, x_0)$ is given by the value of x_0 for which the cost $v_0(1, x_0)$ is a minimum. Hence, from equation 2.22, s_0 is detected as the data element value which is closest to r_0. The latest N elements of the vector $\underline{T}_0(1, s_0')$, are then retained for use in the next cycle of the algorithm, where s_0' is the detected value of s_0. Hence the vector $\underline{Q}_0(1)$ is given by

$$\underline{Q}_0(1) = (-m+1, -m+1, \ldots, -m+1, s_0')$$

and the corresponding cost is given by

$$u_0(1) = (s_0' - r_0)^2 \tag{2.23}$$

This N component vector is extended to the m N+1 component vectors:

$$\underline{T}_1\ (1, x_1) = (-m+1, -m+1, \ldots, -m+1, s_0', x_1)$$

for $x_1 = -m+1, -m+3, \ldots, m-1,$

at the start of the following cycle. The cost for $\underline{T}_1(1, x_1)$ is given by

$$v_1(1,\ x_1) = u_o(1) + \{\underline{Y}.[\underline{Q}_o(1),\ x_1]_{g+1} - r_1\}^2$$

(see equation 2.21). But $\underline{Y}$ is the scalar with value unity so, using equation 2.23,

$$v_1(1,\ x_1) = (s_o' - r_o)^2 + (x_1 - r_1)^2 \tag{2.24}$$

As before, only one of the vectors $\underline{I}_1(1,\ x_1)$ is retained for use in the next cycle of the algorithm, so there.will be only one value of x_1 available for the detection of s_1. The value of x_1, giving the selected vector $\underline{I}_1(1,\ x_1)$, is the value for which $v_1(1,\ x_1)$ is smallest. Hence, from equation 2.24, s_1 is detected as the data element value which is closest to r_1. (This is the data element value for x_1, which minimises $v_1(1,\ x_1)$).

The detection process continues in this way, so that each data element s_i, is detected as the one of its m possible values which is closest to r_i. Therefore, with the ideal channel, the V.A. detector produces the same detected data sequence, as does the optimum combination of linear and decision feedback equalizers. (See sections 1.13 and 1.14). Hence, for this case, the probability of any given data element being in error, is the same as that given in section 1.14.

i.e.

$$P_e = \text{Prob. } (w_i > 1)$$

for a binary signal

and

$$P_e = 1.5 \text{ Prob. } (w_i > 1)$$

for a quaternary signal.

(w_i is, of course, the normally distributed random variable representing Gaussian noise in the system).

CHAPTER 3

3.01 Shortcuts Through the Trellis Diagram

In Section 2.09, it was shown that the Viterbi Algorithm (V.A) requires a large amount of computation, per detected data element, if the sampled impulse response of the channel has many components, or the number of signal levels is fairly large. The number of multiplications plus the number of comparisons, per detected data element, is

$$(2m - 1)\ m^g$$

where m is the number of signal levels and g+1 is the number of components of the sampled impulse response. From Section 2.03 it can be seen that g is determined by the duration of the impulse response, and the time T between successive data elements being transmitted. To transmit information at high speed, T must be small so that many data elements are transmitted per second, or the number m of signal levels must be large. g increases as T decreases so it is clear that

$$(2m - 1)\ m^g$$

may be very large in cases of high speed data transmission. The amount of computation required by the V.A. may then prohibit its use.

The trellis diagram of Figure 2.02 shows the m^g possible values for the vector

$$\underline{Q}_i = (s_{i-g}, s_{i-g+1}, \ldots\ldots s_{i-1})$$

where $\{s_i\}$ is the transmitted data sequence. Using the given definition of the distance between two nodes of the diagram (see Section 2.04), the shortest path from point A to point B, dictates the maximum likelihood sequence $\{s_i\}$.

Suppose that at some stage of the V.A., the shortest path to each of the nodes for some vector $\underline{Q}_i$ have been found, together with the cost function for each of these nodes. The cost function for each node is the length of the shortest path from point A to that node. It therefore seems unlikely that the $\underline{Q}_i$ nodes with relatively large costs, will lie on the shortest path from A to B. The basic principal of the four algorithms described in this chapter, is that of removing all the nodes representing a vector $\underline{Q}_i$, from consideration, except for a fixed number k of them, whose costs are fairly small. Each of these k retained vectors are extended to m new vectors representing $\underline{Q}_{i+1}$, thus giving mk vectors. Then k of these vectors are selected as before, for use in the next cycle of the algorithm.

The four algorithms described below, each have different strategies for deciding which k of the mk nodes available, to select during each cycle. The basic form of the algorithms is the same as that of the V.A., described in Section 2.07. The V.A. holds m^g vectors $\underline{Q}_j(I)$ in store at the beginning of each cycle of the algorithm. m and g are fixed by the data transmission system, so there is no freedom of choice over the number of such vectors used. With Systems 1-4, however, the number of vectors may be chosen to give the desired compromise between the performance of the detection process and its complexity (see Section 3.02). System 1, for example, may be used with any number k of vectors, from one upwards.

As with the V.A., these algorithms may be started off with one vector set equal to

$$(-m+1, -m+1, \ldots., -m+1),$$

or can be started with a full set of k vectors defined in this way. In the latter case, one of the vectors should be assigned a cost equal to zero, and the other vectors given some very large cost (such as 10^6).

3.02 Systems 1-4

These four systems hold a fixed number k, of N component vectors:

$$\underline{Q}_j(1), \underline{Q}_j(2), \ldots\ldots, \underline{Q}_j(k)$$

at the start of a cycle. The number of components of the vectors represents the delay in detecting an element s_i, from the time information about this element first reaches the receiver. Each vector $\underline{Q}_j(I)$ takes the form

$$\underline{Q}_j(I) = (x_{j-N+1}, x_{j-N+2}, \ldots\ldots, x_j) \tag{3.01}$$

where each component x_i has one of the m possible values of the data element s_i. As for the V.A., the transmitted data sequence is denoted

$$\{s_0, s_1, \ldots\ldots, s_n\}$$

and s_i is defined to be equal to -m+1 for $i < 0$ and $i > n$.

From equation 3.01

$$\underline{Q}_{-1}(I) = (x_{-N}, \; x_{-N+1}, \; \ldots\ldots, \; x_{-1})$$

where x_i has one of the possible values of the data element s_i. But s_i = -m+1 for $i < 0$, therefore

$$\underline{Q}_{-1}(I) = (-m+1, \; -m+1, \; \ldots\ldots, \; -m+1)$$

for I = 1, 2, ..., k. (i.e. the k stored vectors $\underline{Q}_{-1}(1)$ are initially all equal to the same vector).

The costs for the vectors $\underline{Q}_{-1}(I)$ are defined by:

$$u_{-1}(I) = \begin{cases} 0 \text{ for } I = 1 \\ \infty \text{ for } I = 2, 3, \ldots., k. \end{cases}$$

The first cycle of the algorithm begins by extending the k vectors $\underline{Q}_{-1}(I)$ to the N+1 component vectors given by

$$\underline{T}_0(I, x_0) = [\underline{Q}_{-1}(I), x_0]$$

for I = 1, 2, ..., k and x_0 taking on the m possible values of a data element. Hence

$$\underline{T}_0(I, x_0) = (-m+1, \; -m+1, \; \ldots\ldots, \; -m+1, \; x_0)$$

for I = 1, 2,....., k

and x_0= -m+1, -m+3, ..., m-1.

The costs corresponding to these extended vectors are given by

$$v_0 \ (I, x_0) = u_{-1}(I) + \{\underline{Y}. \ [\underline{T}_0 \ (I, x_0)]_{g+1} - r_0\}^2$$

where r_0 is the first received signal sample and $\underline{Y}$ is the vector

$$(y_g, y_{g-1}, \ldots\ldots, y_0).$$

(Note that $\underline{Y}$ is the reverse of the vector formed from the channel's sampled impulse response). The term

$$\underline{Y}. \ [\underline{T}_0 \ (I, x_0)]_{g+1}$$

is defined as in Section 2.07.

Then the vector with smallest cost is found, from the set of vectors $\underline{T}_0$ (I, x_0), and the earliest element of this vector is taken as the detected value of s_{-N}. This detected value will be -m+1, as

$$\underline{Q}_{-1}(I) = (-m+1, -m+1, \ldots\ldots, -m+1)$$

for I = 1, 2, ..., k. i.e. all of the vectors $\underline{T}_0$ (I, x_0) have their earliest element equal to -m+1.

All but k of the mk extended vectors are deleted from storage, according to decision rules which are different for each of the four systems. (These decision rules are described below). The latest N elements of the remaining k vectors $\underline{T}_0$ (I, x_0) are then stored in the array $\underline{Q}_0$(J), for J = 1, 2, ..., k. The corresponding costs v_0(I, x_0) are stored in the array u_0(J). This completes the first cycle of the algorithms.

At the start of the j+1 st. cycle, k vectors $\underline{Q}_{j-1}(I)$ are stored, together with their costs $u_{j-1}(I)$. These k vectors are extended to the N+1 component vectors $\underline{T}_j(I, x_j)$, by the addition of the component x_j, which can take on m values. These mk extended vectors are given by

$$\underline{T}_j(I, x_j) = [\underline{Q}_{j-1}(I), x_j] \qquad (3.02)$$

The corresponding costs are given by

$$v_j(I, x_j) = u_{j-1}(I) + \{\underline{Y}\cdot[\underline{Q}_{j-1}(I), x_j]_{g+1} - r_j\}^2 \qquad (3.03)$$

where r_j is the (j+1)st received signal sample.

The element s_{j-N} is detected as the earliest element of the vector, from the set $\underline{T}_j(I, x_j)$, which has smallest cost. Then all but k of the extended vectors $\underline{T}_j(I, x_j)$ are deleted, and the remaining k vectors are stored in the array $\underline{Q}_j(J)$, for J = 1, 2, ..., k. (In fact only the latest N elements are stored). The corresponding costs $v_j(I, x_j)$, for the k selected vectors, are stored in the array $u_j(J)$. This completes the j+1 st. cycle of the process. The algorithm continues in this way until the entire data sequence $\{s_i\}$ has been detected. A flow diagram for Systems 1-4 is given in Figure 3.01.

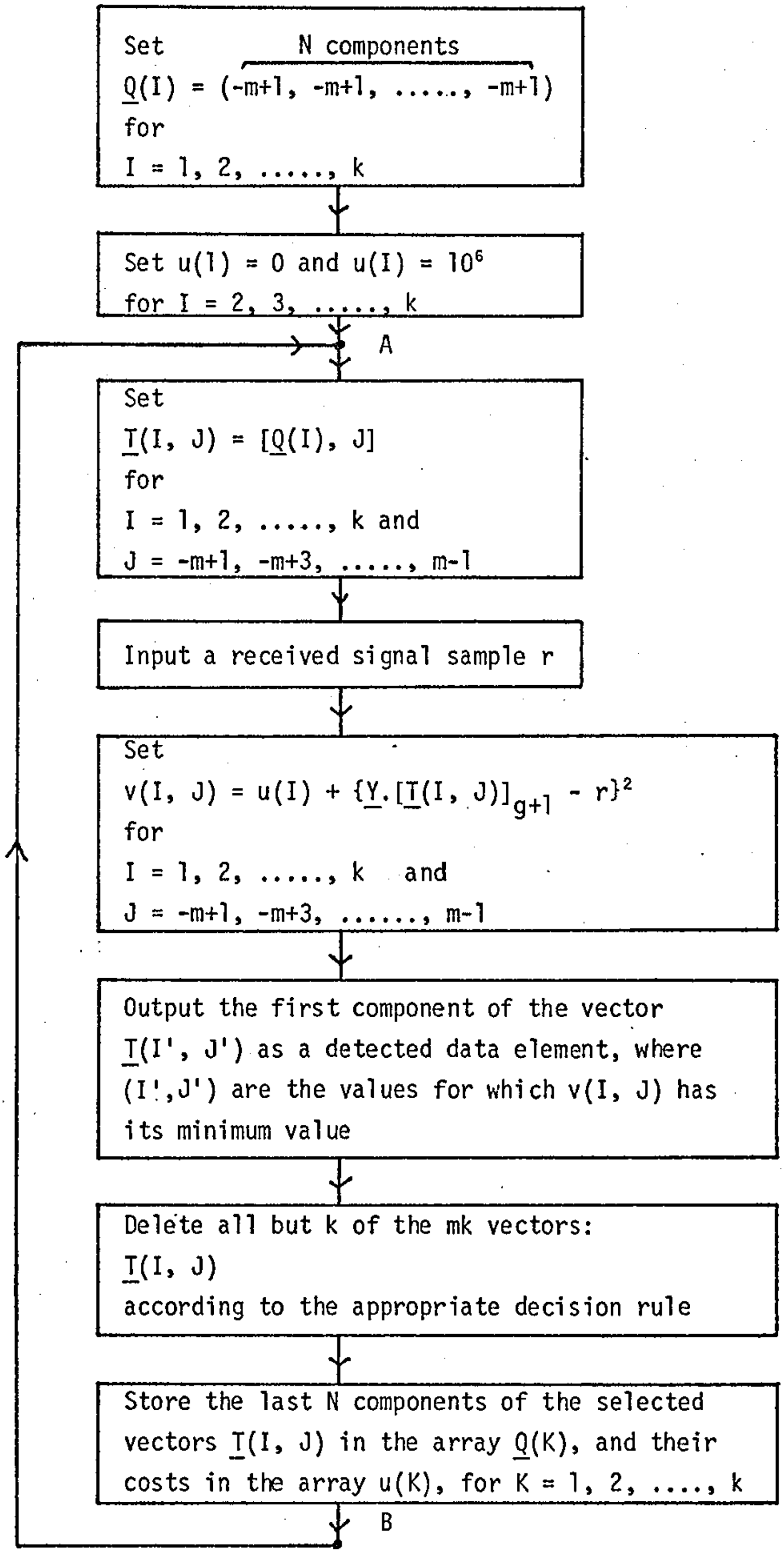

Set $\underline{Q}(I) = (\underbrace{-m+1, -m+1, \ldots\ldots, -m+1}_{N \text{ components}})$ for $I = 1, 2, \ldots\ldots, k$
Set $u(1) = 0$ and $u(I) = 10^6$ for $I = 2, 3, \ldots\ldots, k$
A
Set $\underline{T}(I, J) = [\underline{Q}(I), J]$ for $I = 1, 2, \ldots\ldots, k$ and $J = -m+1, -m+3, \ldots\ldots, m-1$
Input a received signal sample r
Set $v(I, J) = u(I) + \{\underline{Y}.[\underline{T}(I, J)]_{g+1} - r\}^2$ for $I = 1, 2, \ldots\ldots, k$ and $J = -m+1, -m+3, \ldots\ldots, m-1$
Output the first component of the vector $\underline{T}(I', J')$ as a detected data element, where (I',J') are the values for which $v(I, J)$ has its minimum value
Delete all but k of the mk vectors: $\underline{T}(I, J)$ according to the appropriate decision rule
Store the last N components of the selected vectors $\underline{T}(I, J)$ in the array $\underline{Q}(K)$, and their costs in the array $u(K)$, for $K = 1, 2, \ldots\ldots, k$
B

FIGURE 3.01

Block diagram for Systems 1-4

3.03 Decision Rules for Systems 1-4

Each of the systems 1-4 works in a similar manner, except for the decision rules which dictate which vectors are to be rejected during each cycle. k N component vectors are stored at the start of each cycle of the algorithms. These vectors are then extended to mk N+1 component vectors, by the addition of another component with m possible values, to each of them. A data element is then detected and all but k of the mk vectors are deleted from storage. The rules which decide which k vectors to retain are described below, for each of the systems.

Algorithms similar to the one employing decision rule 1, have been proposed independently by F.L. Vermeulen, S.A. Fredricsson, G.J. Foschini, J. Gordon and N. Montague [40-43]. The decision rules for Systems 2, 3 and 4 are due to A.P. Clark.

3.04 Decision Rule 1 (System 1)

This decision rule is the simplest of the four. With it, System 1 selects the k vectors with smallest costs from the set of mk extended vectors, during each cycle of the algorithm.

3.05 Decision Rule 2 (System 2)

For System 2, the number k of vectors stored at the start of each cycle of the algorithm, must be a multiple of the number m of signal levels.

Let $\ell = k/m$.

Then the number of vectors stored at the start of each cycle is $m\ell$. These $m\ell$ N component vectors are extended to $m^2\ell$ N+1 component vectors, as explained in Section 3.02. These extended vectors are divided into m groups, with the vectors in each group all having the same latest component. Then the ℓ vectors with smallest costs are retained from each group, giving ℓm (or k) stored vectors again. This rule ensures that, for each possible value of the latest component of the vectors, there will be an equal number of vectors in the system, having this latest component. (The latest component being the one furthest to the right in the vectors).

One foreseeable problem with System 1, is that it is possible for all of the k vectors selected by decision rule 1, to have the same latest component x_j. If this is the case, then there is only one possible value available for the detection of s_j, at a later stage of the algorithm. The algorithm will then have inadvertently made a decision on a data element even though only the first sample r_j, containing information about this data element, had been received. For cases where the first element of the channel's sampled impulse response is small, this first sample will contain only a small amount of information about the data element s_j. (See equation 2.07). A detection of this type is clearly undesirable. It is not easy to predict the seriousnessof this factor in System 1, without performing simulation tests. However, the decision rule for System 2 overcomes any difficulty, that may arise from this possible disadvantage of System 1.

3.06 Decision Rule 3 (System 3)

As for System 2, the number k of vectors stored at the start of each cycle, must be a multiple of m for System 3.

Let

$\ell = k/m$ as before.

The vectors stored at the start of the (j+2)nd cycle are denoted by

$$\underline{Q}_j(I) = (x_{j-N+1}, x_{j-N+2}, \ldots\ldots, x_j)$$

for I = 1, 2, ..., ℓm. Decision rule 3 ensures that these vectors contain all possible values of the latest ℓ elements:

$$x_{j-\ell+1}, x_{j-\ell+2}, \ldots\ldots, x_j.$$

except while the process is starting up. From the expanded set of vectors of the form:

$$\underline{T}_{j+1}\,(I, x_{j+1}) = (x_{j-N+1}, x_{j-N+2}, \ldots\ldots, x_{j+1}),$$

ℓm vectors must be selected and retained for use in the following cycle of the algorithm. Decision rule 3 selects these ℓm vectors as follows:

Some particular value for $x_{j-\ell+2}$ is chosen and, from the set of vectors which have this value for $x_{j-\ell+2}$, the one with smallest cost is selected. This is repeated for the other m-1 possible values of $x_{j-\ell+2}$, giving m selected vectors so far. In the same manner, m vectors are selected corresponding to the m possible values of

$$x_{j-\ell+3}, x_{j-\ell+4}, \ldots\ldots, x_{j+1}$$

giving a total of ℓm vectors. One restriction on the selection procedure is that no vector may be selected twice.

ℓm vectors are selected corresponding to the different possible values of the components

$$x_{j-\ell+2},\ x_{j-\ell+3},\ \ldots\ldots,\ x_{j+1}$$

so the stored vectors $\underline{Q}_j(I)$ must have at least ℓ-1 components. Hence, if it is decided that some particular value of N is to be used with System 3, then the maximum value for ℓ is N+1. m(N+1) is then the maximum number of vectors that may be stored at the start of each cycle of the algorithm.

Note that, when the detection process is starting up, all k of the stored vectors are set equal to the N component vector

$$(-m+1,\ -m+1,\ \ldots\ldots,\ -m+1)$$

(see Section 3.02). Until several cycles of the algorithm have been completed, the earlier components of all of the stored vectors will be equal to -m+1. Therefore it will not be possible for a set of k vectors, with all possible element values in the latest ℓ components, to be selected. Hence decision rule 3 needs to be modified for the first few cycles of the algorithm. The modified rule chooses vectors with a full selection of element values, in as many as possible of the latest components. The set of k vectors is then completed with an arbitrary selection from the remaining vectors.

Example

Consider a case with

$$N = 3$$

$$m = 2 \quad \text{and}$$

$$k = 6$$

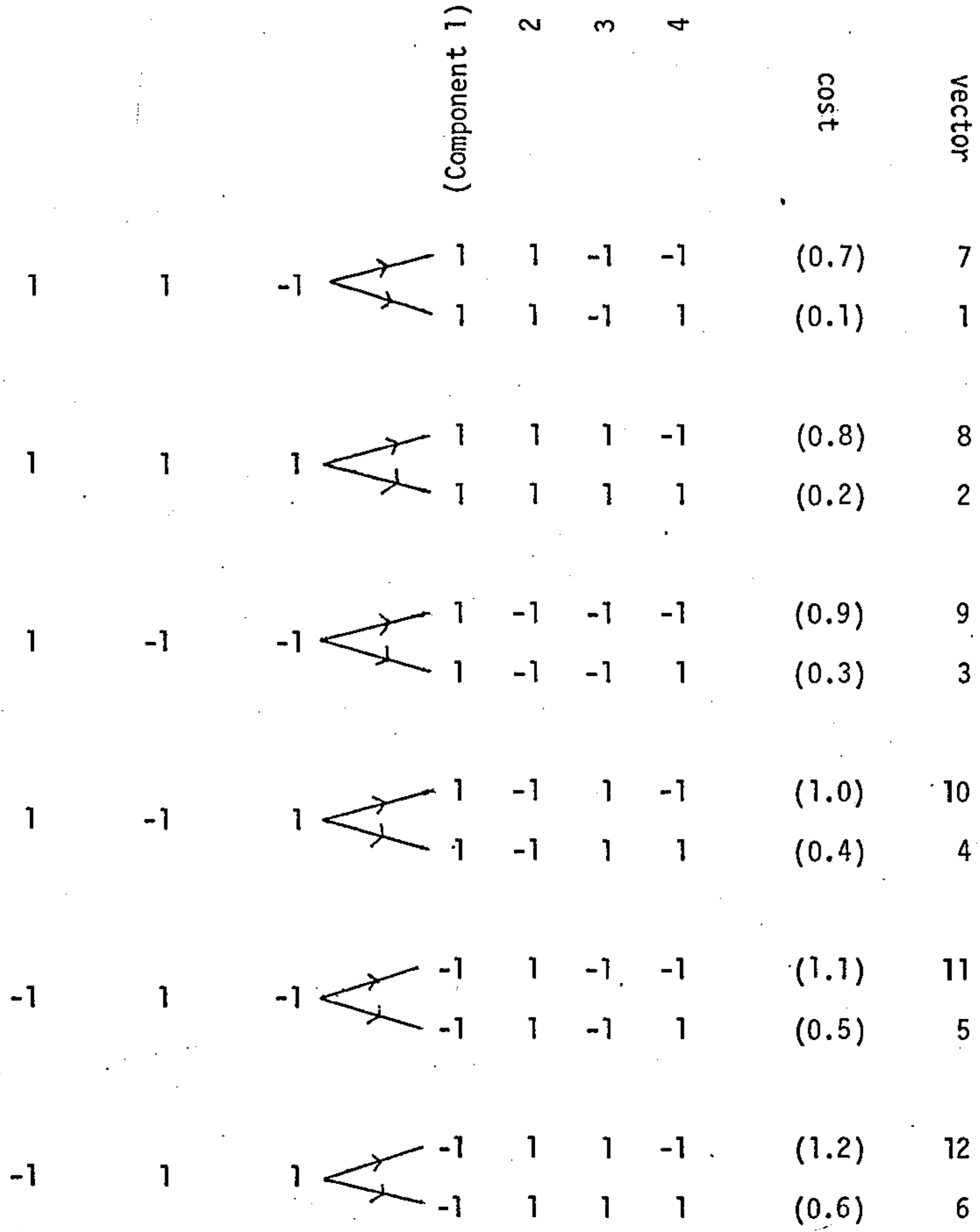
i.e. a two level signal with 6 vectors stored at the start of each cycle of the algorithm, the vectors each having three components. Let the 6 vectors in store, at the start of some cycle of the process, be those on the left in Figure 3.02. These six vectors are extended to twelve four component vectors, by adding either a 1 or a -1 to the right hand side of each, as shown. Let the costs for the extended vectors be those given in brackets in Figure 3.02.

ℓ is defined equal to k/m, so

$$\ell = 6/2 = 3.$$

Hence, decision rule 3 gives a variety of element values in the latest 3 components of the vectors (i.e. in the 3 components furthest to the right of the selected vectors). Let the components of the extended vectors be denoted 1, 2, 3 and 4, as shown, with the vectors denoted 1, 2, ..., 12.

The first step of the decision rule is to select the vector with smallest cost, which has component 2 equal to -1. Hence vector 3 is selected. Then the vector with smallest cost, which has component 2 equal to 1 is selected. This is vector 1. These vectors are now removed from consideration so that they may not be selected again. In a similar manner, two vectors are selected which have component 3



			(Component 1)	2	3	4	cost	vector
1	1	-1	1	1	-1	-1	(0.7)	7
			1	1	-1	1	(0.1)	1
1	1	1	1	1	1	-1	(0.8)	8
			1	1	1	1	(0.2)	2
1	-1	-1	1	-1	-1	-1	(0.9)	9
			1	-1	-1	1	(0.3)	3
1	-1	1	1	-1	1	-1	(1.0)	10
			1	-1	1	1	(0.4)	4
-1	1	-1	-1	1	-1	-1	(1.1)	11
			-1	1	-1	1	(0.5)	5
-1	1	1	-1	1	1	-1	(1.2)	12
			-1	1	1	1	(0.6)	6

FIGURE 3.02

Selection of vectors by decision rule 3

equal to -1 and component 3 equal to 1. The ones with lowest costs are vectors 5 and 2. Finally vectors 4 and 7 are selected, to give a set of 6 vectors.

Now consider again the possible difficulty mentioned for System 1, that the set of k vectors selected during some cycle, may all have the same latest component. System 2 ensures that such a set of vectors will not be selected, but it is still possible that the selected vectors may have one of their latest elements in common. System 3 should offer more protection against this sort of difficulty, as it ensures a variety of element values in the latest ℓ components of the vectors.

3.07 Decision Rule 4 (System 4)

For System 4, the number k of vectors stored at the start of each cycle, must be such that

$$k = m^{\ell}$$

for some positive integer ℓ (where m is the number of signal levels). Rule 4 ensures that these vectors contain all possible combinations of values, in their latest ℓ components (except while the process is starting up). Hence, if m^{ℓ} vectors are stored at the start of each cycle, the number N of components of these vectors must be greater than or equal to ℓ. Alternatively, if it is decided that a certain value of N is to be used with System 4, the maximum value for k is m^N.

Note that, with its decision rule, System 4 also ensures that premature detections will not be made due to the selected vectors having any of their latest ℓ elements in common.

During each cycle of the algorithm, k of the mk extended vectors must be selected for use in the following cycle. System 4 is such that these k (or m^ℓ) vectors must contain all possible combinations of the latest ℓ components. Hence, for each of these combinations, the vector with lowest cost is selected, giving m^ℓ vectors in all. (There are m^ℓ possible combinations of the latest ℓ elements, as each element has m possible values).

For the first few cycles of the algorithm, it will not be possible to select a set of vectors with all possible combinations of the latest ℓ elements. This is because all of the vectors are initially given the same components, i.e. all components are set equal to -m+1. Hence, as with System 3, the selection procedure must be modified for the first few cycles of the algorithm. For the first few cycles, the vector with smallest cost is selected, for all combinations of as many as possible of the latest components of the vectors. The set of m^ℓ selected vectors is then completed by choosing vectors arbitrarily from those remaining.

Note that System 4 is identical to the V.A. detector if ℓ is set equal to g.

3.08 Starting up Procedure

As with the V.A., the data elements s_i are defined to be equal to -m+1, for $i < 0$ and $i > n$ (see Section 2.08). Hence the transmitter must send out a training signal of N elements with the value -m+1, before transmitting the data sequence:

$s_0, s_1, \ldots\ldots, s_n.$

(N is, of course, the number of components of the vectors $\underline{Q}_j(I)$).

It has been found that the V.A. will synchronize itself to the correct data sequence, after a few cycles, even if an incorrect training signal is used [33]. It seems likely that Systems 1-4 will also have this property, and, apart from an initial burst of errors, will function correctly with a false training sequence.

There is one particular start up procedure for Systems 1 and 2, which gives a very poor performance. Suppose that all of the k vectors of System 1, are set equal to the same vector, and that their costs also have the same value. In particular, suppose that all of the vectors $\underline{Q}_{-1}$ (I) are set equal to (-m+1, -m+1, ... -m+1), and all of the costs $u_{-1}(I)$ are set equal to zero. Then the expanded set of vectors is given by

$$\underline{T}_0(I, x_0) = (-m+1, -m+1, \ldots\ldots, -m+1, x_0)$$

with costs:

$$u_0(I, x_0) = u_{-1}(I) + \{\underline{Y}.[\underline{T}_0(I, x_0)]_{g+1} - r_0\}^2$$

$$= 0 \quad + \quad \{\underline{Y}.[\underline{T}_0(I, x_0)]_{g+1} - r_0\}^2$$

(see Section 3.02). Therefore the values of the vectors $\underline{T}_0(I, x_0)$, and their costs $u_0(I, x_0)$ are independent of I.

Hence the vectors split into m groups, according to the value of x_0, with all vectors in a group being identical and all costs being identical. Now assume that the m costs for the different groups are distinct. (This assumption is supported by simulation results). Then, from the set of mk costs for the extended vectors, the k smallest costs will come from the same group. Hence decision rule 1, will select k vectors from the same group, and the selected vectors will be identical, and will all have identical costs. The situation with k identical vectors and k identical costs, at the start of the first cycle, will be preserved at the start of the second cycle. It can be seen, therefore, that each following cycle of the process will begin with k identical vectors, and System 1 is effectively functioning with k=1. The ability of the detection process to store a reasonable number of possible data sequences, has then been lost and the performance of the detector may be reduced.

The recommended starting up procedure is given in Section 3.02. If this procedure is followed, one vector is given a cost equal to zero, and the other vectors are given very large costs. It can then be shown that a distinct set of vectors will be present in the system, after a few cycles. (See the proof of theorem 3.02 in Section 3.10).

The starting procedure with all vectors identical and having one common cost, is also fairly disastrous for System 2. Suppose that System 2 is initialized by setting the k vectors $\underline{Q}_{-1}(I)$ equal to

$$(-m+1, -m+1, \ldots\ldots, -m+1)$$

and setting the k costs equal to zero. The expanded set of vectors will then be given by

$$\underline{T}_0(I, x_0) = (-m+1, -m+1, \ldots\ldots, -m+1, x_0)$$

with costs given by

$$u_0(I, x_0) = u_{-1}(I) + \{\underline{Y}.[\underline{T}_0(I, x_0)]_{g+1} - r_0\}^2$$

$$= 0 \quad + \quad \{\underline{Y}.[\underline{T}_0(I, x_0)]_{g+1} - r_0\}^2$$

(see Section 3.02). As for System 1, the vectors form m groups with k identical vectors in each group. The vectors in each group also have a common cost. It will be assumed that the costs for each group are all different. The decision rule for System 2 will select the k/m vectors with smallest cost, for each possible value of x_0. Hence, k/m vectors will be selected from each group. Then, at the start of the second cycle of the algorithm, the k vectors $\underline{Q}_0(I)$ form m groups of vectors, with the vectors in each group being identical, and having a common cost.

Theorem 3.01

Suppose that, at the start of some cycle of System 2, the vectors stored may be divided into m groups, with the k/m vectors in each group being identical. Assume also that the vectors in any one group have a common cost, but that the costs are different from one group to another. (m is, of course, the number of signal levels). Then this grouping of vectors and costs, will be maintained at the start of the following cycle of the algorithm .

Proof

Let the set of ℓm (or k) vectors, stored at the start of some cycle of the algorithm, be

$$\underline{v}_{1,1},\ \underline{v}_{1,2},\ \ldots\ldots,\ \underline{v}_{1,\ell} \qquad \text{with costs} = c_1$$

$$\underline{v}_{2,1},\ \underline{v}_{2,2},\ \ldots\ldots,\ \underline{v}_{2,\ell} \qquad \text{with costs} = c_2$$

$$\vdots$$

$$\underline{v}_{m,1},\ \underline{v}_{m,2},\ \ldots\ldots,\ \underline{v}_{m,\ell} \qquad \text{with costs} = c_m.$$

Then the j th. vector in the group with cost c_i, is denoted $\underline{v}_{i,j}$. The vectors in each group are the same, so

$$\underline{v}_{ij} = \underline{v}_{ik} \qquad (3.04)$$

for i = 1, 2, ..., m, j = 1, 2, ..., ℓ and k = 1, 2, ...,ℓ.

These vectors are extended to the $m^2\ell$ vectors $(\underline{v}_{ij}, x)$, by the addition of a new element x which can take on m values. The cost for the vector $(\underline{v}_{ij}, x)$ is given by

$$D(i,x) = c_i + \{\underline{Y}.[\underline{v}_{i,j}, x]_{g+1} - r\}^2$$

for i = 1, 2, ..., m and x = -m+1, -m+3, ... m-1, where r is the appropriate received signal sample. (See Section 3.02). From equation 3.04, it can be seen that this cost is independent of j, and for each pair of values (i,x), there are ℓ vectors with cost D(i,x). Assume that the costs D(i,x) are all different, for different values of i and x. (This is supported by simulation results).

Decision rule 2 selects the ℓ vectors with smallest cost, for each possible value of the latest component x. Let i(x) be the value of i which minimises D(i,x), for a given value of x. Then, for given x, the ℓ vectors with smallest costs are

$$(\underline{v}_{i(x),1}, x), \quad (\underline{v}_{i(x),2}, x), \; \ldots\ldots, \; (\underline{v}_{i(x),\ell}, x).$$

which all have cost equal to D[i(x), x].
Hence, the mℓ vectors selected by System 2 are

$$(\underline{v}_{i(x),1}, x), \quad (\underline{v}_{i(x),2}, x), \; \ldots\ldots,(\underline{v}_{i(x),\ell}, x)$$

for x = -m+1, -m+3, ..., m-1. But, from equation 3.04,

$$v_{i(x),1} = v_{i(x),2} = \ldots\ldots = v_{i(x),\ell}$$

for any value of x. Hence the mℓ selected vectors divide into m groups, according to their value of x, with the vectors in any one group being all the same. The vectors in any group also have the common cost D[i(x), x].
End of proof.

If System 2 is started up with all of its vectors set equal to the same vector, and all having a common cost, they become grouped in the form indicated in theorem 3.01. This grouping will then be maintained throughout the operation of System 2, except possibly if the costs for two groups become equal. Simulation results show that this is a rare or impossible event, so System 2 would then be stuck in a mode of operation in which only m of its vectors were distinct, i.e. the number of vectors stored, at the start of each cycle of the algorithm, would be effectively reduced from $m\ell$ to m.

The decision rules for System 3 and 4 ensure that a reasonable variety of vectors are always present in the algorithm. Therefore these systems will never enter a mode of operation, in which the number of stored vectors is effectively reduced, as can happen with Systems 1 and 2 under certain conditions. In fact, System 4 selects vectors in such a way, that all combinations of element values are present, in as many of the latest components of the vectors as is possible. This ensures that all of the vectors stored by System 4 will be distinct, except for the first few cycles of the algorithm.

3.09 Number of Operations Required by Systems 1-4

System 1

System 1 selects the k vectors with smallest cost, from a group of mk vectors, during each cycle of the algorithm. Hence the k smallest costs must be found, from a group of mk costs. There are many methods of varying efficiency, for solving this problem, but a fairly simple method will be assumed here. Let the k smallest costs be selected as follows:

First select the smallest cost from the group of mk. This requires mk-1 comparisons between two numbers. The selected cost is then removed from consideration, and the process repeated to find the smallest cost from a group of mk-1. This requires mk-2 comparisons. In this manner, the k smallest costs may be found with a number of comparisons equal to

$$(mk-1) + (mk-2) + \ldots\ldots + (mk-k)$$

$$= mk^2 - \tfrac{1}{2}k\ (k+1)$$

System 2

With System 2, the expanded set of mk vectors is considered in m separate groups of k, each group having a common value for the latest component of its vectors. The ℓ (or k/m) vectors with smallest cost must then be selected from each group of k. Hence using the procedure described above for System 1, the number of comparisons needed for each group is

$$(k-1) + (k-2) + \ldots\ldots + (k - k/m)$$

$$= k^2/m - \frac{k}{2m}\ \left(\frac{k}{m} + 1\right)$$

There are m groups, so the total number of comparisons required for a cycle of System 2 is

$$m\left(\frac{k^2}{m} - \frac{k}{2m}\ \left(\frac{k}{m} + 1\right)\right)$$

or

$$k^2 - \frac{k}{2}\ \left(\frac{k}{m} + 1\right).$$

System 3

Let $\ell = k/m$ as in Section 3.06. System 3 works by first considering the set of mk expanded vectors in m separate groups. The groups are divided according to the value of the component which is the ℓth from the right of the vectors. The vector with lowest cost is then selected from each group, and removed from consideration for the remainder of the selection process.

Let the number of vectors in each of these groups be:

$$n_1, n_2, \ldots\ldots, n_m$$

where

$$n_1 + n_2 + \ldots\ldots + n_m = mk \qquad (3.05)$$

The number of comparisons required to find the vector with smallest cost, from a group of n_i vectors, is $n_i - 1$. Hence the number of comparisons required to select one vector from each of the m groups is:

$$(n_1 - 1) + (n_2 - 1) + \ldots\ldots + (n_m - 1)$$

$$= n_1 + n_2 + \ldots\ldots + n_m - m$$

$$= mk - m$$

(using equation 3.05). Then m vectors have been selected, and there are still mk - m vectors available for selection. These mk - m vectors are now split into m groups, according to the values they have for the $(\ell-1)$st element from the right. Note that, although every possible value of this element may be present in the larger set of mk vectors, some values may not be available in the remaining set of mk-m vectors. Hence some of the m groups, formed according to the value of the component which is (ℓ-1)st from the right, may be empty. However many simulation tests have been performed with System 3, and this situation was never found to arise. It will therefore be assumed, for this section, that each of the m groups has at least one vector.

It has been shown above that, for the selection of the first m vectors from a set of mk, mk-m comparisons are required by System 3. Similarly, for the selection of the next m vectors from a set of mk-m, the number of comparisons required is

$$(mk-m) - m$$

or $m(k-2)$

The remaining mk-2m vectors are then split into m groups, according to the values they have for the component which is (ℓ-2)nd from the right, and so on. Then the total number of comparisons required for the selection of the k vectors is

$$m(k-1) + m(k-2) + \ldots\ldots + m(k-\ell)$$

$$= mk\ell - \tfrac{1}{2}m\ell(\ell+1)$$

But ℓ is defined equal to k/m, so the required number of comparisons per cycle of the algorithm is

$$k^2 - \tfrac{1}{2}k\left(\frac{k}{m} + 1\right)$$

which is the same as for System 2.

System 4

Let ℓ be an integer such that

$$m^{\ell} = k$$

as in Section 3.07. System 4 functions by considering the mk extended vectors in m^{ℓ} separate groups, corresponding to the m^{ℓ} possible combinations, of the latest ℓ components of the vectors. Each of these groups contains m vectors, and the vector with smallest cost must be selected from each group. Hence m-1 comparisons must be made, for each of the m^{ℓ} (or k) groups and the number of comparisons required, per cycle of the algorithm, is

$$k\ (m-1).$$

From Section 3.02, it can be seen that mk multiplications (or squaring operations) must be performed by Systems 1-4, for the calculation of the costs during each cycle.

Now consider a particular situation, in which a two level signal is transmitted over a channel whose sampled impulse response has fifteen components.

Then $m = 2$

and $g = 14$.

Simulation results have shown that a value for k of 16, was sufficient for Systems 1-4 to give a performance close to that of the V.A. detector, for some such channels. Hence k will be taken to be 16, for this example.

Table 3.01 gives the number of multiplications required per cycle, for the V.A. detector and Systems 1-4, for this situation.

(The number of operations required by the V.A. was derived in Section 2.09). It can be seen from the table, that each of the Systems 1-4 offers a large saving in the number of basic operations, over the V.A. detector. System 4 offers the greatest saving, and the number of multiplications and comparisons required, is only one hundredth of that required by the V.A. detector.

Detection Processes	Number of multiplications and comparisons required for each element detected, with m = 2, k = 16, and g = 14
V.A.	$m^{g+1} + (m-1)\,m^{g} = 49152$
System 1	$mk + mk^2 - \frac{1}{2}\,k(k+1) = 408$
System 2)	$mk + k^2 - \frac{k}{2}\,(\frac{k}{m} + 1) = 216$
System 3)	
System 4	$mk + k\,(m-1) = 48$

TABLE 3.01

3.10 The Effect of a Zero as the First Component of the Channel's Sampled Impulse Response

Consider the case of a transmission channel whose characteristics vary with time, so that channel vector (or sampled impulse response), is not constant. Assume also that the detector stores an estimate of the channel's sampled impulse response, which has a fixed number of components. Then some device for estimating the channel vector must be used in conjunction with the detector. Now, in a situation where the duration of the channel's impulse response varies with time, some components of the estimated channel vector may be set to zero.

Suppose that the sampled impulse response of the channel under consideration is

$$(y_1, y_2, \ldots\ldots, y_g)$$

Define System A to be System 1 with the channel vector estimated correctly as

$$(y_1, y_2, \ldots\ldots, y_g)$$

and let System B be System 1 with the channel vector estimated as

$$(y_0, y_1, \ldots\ldots, y_g)$$

where $y_0 = 0$.

Let System A have k N component vectors stored at the start of each cycle, and let System B have mk N+1 component vectors stored at the start of each cycle. It will now be shown that Systems A and B are equivalent. (This result and similar results for Systems 2 and 4 were suggested by A P Clark).

Lemma 3.01

Let the k vectors of System A, at the start of some cycle of the algorithm be denoted $\underline{Q}(I)$, with a distinct set of costs u(I), for

$$I = 1, 2, \ldots\ldots, k.$$

Let the signal sample received during this cycle be r.

Now suppose that the mk vectors of System B, at the start of the cycle in which r is received, are of the form

$(\underline{Q}\ (I),\ J)$

for

$I = 1, 2, \ldots.., k$ and

$J = -m+1, -m+3, \ldots.., m-1.$

(J takes on the m possible values of a data element). Suppose also that the costs for the vectors

$(\underline{Q}(I),\ J)$

are independent of J and equal to

$u(I) + c$

for some constant c.

Then this relationship between the vectors and costs of Systems A and B, will be maintained at the start of the following cycle of the algorithms.

Proof

For System A, the extended set of mk vectors is

$(\underline{Q}(I),\ K)$

for

$I = 1, 2, \ldots.., k$ and

$K = -m+1, -m+3, \ldots.., m-1.$

The costs for these vectors are given by:

$$v(I,K) = u(I) + \{\underline{Y}^A.\ [\underline{Q}(I),\ K]_g - r\}^2$$

where

$$\underline{Y}^A = (y_g, y_{g-1}, \ldots\ldots, y_1)$$

(see Section 3.02).

Let the k pairs (I, K) which give the k smallest values of v(I, K) be denoted

$$(I_\ell, K_\ell)$$

for $\ell = 1, 2, \ldots\ldots, k$.

Then the k vectors selected by System A are

$$[\underline{Q}(I_\ell), K_\ell] \quad \text{with costs} \quad v[I_\ell, K_\ell]$$

for $\ell = 1, 2, \ldots., k$.

For System B, the extended set of m^2k vectors is

$$[\underline{Q}(I), J, L] \quad \text{for}$$

$$I = 1, 2, \ldots\ldots, k$$

$$J = -m+1, -m+3, \ldots\ldots, m-1 \quad \text{and}$$

$$L = -m+1, -m+3, \ldots\ldots, m-1.$$

The costs for these vectors are given by

$$D(I, J, L) = u(I) + c + \{\underline{Y}^B. [\underline{Q}(I), J, L]_{g+1} - r\}^2 \qquad (3.07)$$

where:

$$\underline{Y}^B = (y_g, y_{g-1}, \ldots\ldots, y_1, 0)$$

(see Section 3.02).

$$\underline{Y}^B . [\underline{Q}(I), J, L]_{g+1}$$

is the scalar product of $\underline{Y}^B$ and the latest g+1 components of $[\underline{Q}(I), J, L]$ (i.e. the g+1 components furthest to the right). But, from the definitions of $\underline{Y}^A$ and $\underline{Y}^B$, it is clear that

$$\underline{Y}^B = (\underline{Y}^A, 0)$$

$$\therefore \; \underline{Y}^B . [\underline{Q}(I), J, L]_{g+1} = \underline{Y}^A . [Q(I), J]_g$$

Hence, from equation 3.07,

$$D(I, J, L) = u(I) + c + \{\underline{Y}^A . [\underline{Q}(I), J]_g - r\}^2 \qquad (3.08)$$

Comparing equations 3.06 and 3.08 gives

$$D(I, J, L) = c + v(I, J) \qquad (3.09)$$

Now the k pairs (I, J) which minimise v(I, J) have been denoted

$$(I_\ell, K_\ell)$$

for $\ell = 1, 2, \ldots\ldots, k$

From equation 3.09, the value of D(I, J, L) is independent of L. Hence the mk values of (I, J, L) giving the smallest values of D(I, J, L) are

$$[I_\ell, K_\ell, L]$$

for $\ell = 1, 2, \ldots\ldots, k$ and

$$L = -m+1, -m+3, \ldots\ldots, m-1.$$

The mk vectors selected by System B are therefore:

$[\underline{Q}(I_\ell), K_\ell, L]$ for

$\ell = 1, 2, \ldots., k$ and

$L = -m+1, -m+3, \ldots.., m-1.$

From equation 3.09, the corresponding costs for these mk vectors are given by

$$D[I_\ell, K_\ell, L] = c + v[I_\ell, K_\ell] \quad .$$

These are the same as the costs for the vectors

$[\underline{Q}(I_\ell), K_\ell]$

selected by System A, except for the additive constant c. Hence the relationship between the vectors and costs of Systems A and B, will be preserved at the start of the following cycle of the algorithm.

End of proof.

Theorem 3.02

The Systems A and B, defined above, will produce the same detected data sequences.

Proof

The theorem will be proved with the help of lemma 3.01.

System A, 1st Cycle:

System A initially has k stored vectors equal to the N component vector

$$(-m+1,\ -m+1,\ \ldots\ldots,\ -m+1)$$

One of these vectors is assigned a zero cost and the others, an infinite cost. The extended set of mk vectors contains m vectors of the form

$$(-m+1,\ -m+1,\ \ldots\ldots,\ -m+1,\ I_1)$$

with costs

$$\{\underline{Y}^A.\ [-m+1,\ -m+1,\ \ldots\ldots,\ -m+1,\ I_1]_g - r_1\}^2$$

where

$$I_1 = -m+1,\ -m+3,\ \ldots\ldots,\ m-1$$

(see Section 3.02). The set of expanded vectors also contains m(k-1) vectors with infinite costs. r_1 is the first signal sample received by System A.

The decision rule for System A (decision rule 1) now selects the k vectors with smallest costs, from the set of mk vectors. Hence the k selected vectors will contain m vectors of the form

$$(-m+1,\ -m+1,\ \ldots\ldots,\ -m+1,\ I_1),$$

with costs

$$\{\underline{Y}^A.\ [-m+1,\ -m+1,\ \ldots\ldots,\ -m+1,\ I_1]_g - r_1\}^2$$

and k-m vectors with costs = ∞ (assuming that $k \geq m$).

System A, 2nd Cycle

The set of mk expanded vectors will contain m^2 vectors of the form

$$(-m+1, -m+1, \ldots\ldots, -m+1, I_1, I_2)$$

with costs

$$\{\underline{Y}^A.\ [-m+1, -m+1, \ldots\ldots, -m+1, I_1]_g - r_1\}^2$$

$$+ \{\underline{Y}^A.\ [-m+1, -m+1, \ldots\ldots, -m+1, I_1, I_2]_g - r_2\}^2$$

(see Section 3.02), where r_2 is the second signal sample received by System A.

I_1 and I_2 may each take on the values

$$-m+1, -m+3, \ldots\ldots, m-1.$$

The setof expanded vectors will also contain m(k-m) vectors with costs = ∞.

Hence the k selected vectors (the ones with lowest costs) will contain m^2 vectors of the form

$$(-m+1, -m+1, \ldots\ldots, -m+1, I_1, I_2)$$

with costs

$$\sum_{j=1}^{2} \{\underline{Y}^A.\ [-m+1, -m+1, \ldots., -m+1, I_1, I_2, \ldots\ldots, I_j]_g - r_j\}^2$$

The set of selected vectors will also contain $k-m^2$ vectors with infinite costs (assuming that $k \geq m^2$).

System A, i^{th} cycle (where i is the smallest integer such that $m^i \geq k$):

The expanded set of mk vectors will contain m^i vectors of the form

$$(-m+1, -m+1, \ldots\ldots, -m+1, I_1, I_2, \ldots\ldots, I_i)$$

with costs

$$\sum_{j=1}^{i} \{\underline{Y}^A . [-m+1, -m+1, \ldots\ldots, -m+1, I_1, I_2, \ldots\ldots, I_j]_g - r_j\}^2$$

where each I_j may take on the m values

$$-m+1, -m+3, \ldots\ldots, m-1$$

and r_j is the jth signal sample received by System A. The expanded set of vectors will also contain $mk-m^i$ vectors with costs = ∞.

i has been defined to be the smallest integer such that $m^i \geq k$, so there are now k vectors with finite costs, present in the set of expanded vectors. Hence the k selected vectors will be ones with finite costs.

Let

$$[I_1(\ell), I_2(\ell), \ldots\ldots, I_i(\ell)]$$

be the values of

$$[I_1, I_2, \ldots\ldots, I_i]$$

which give the k smallest values for the cost function

$$\sum_{j=1}^{i} \{\underline{Y}^A . [-m+1, -m+1, \ldots\ldots, -m+1, I_1, I_2, \ldots\ldots, I_j]_g - r_j\}^2$$

where ℓ may take on the values 1, 2, ..., k. Then the k selected vectors are the N+1 component vectors

$$[-m+1, -m+1, \ldots\ldots, -m+1, I_1(\ell), I_2(\ell), \ldots\ldots, I_i(\ell)]$$

for

$$\ell = 1, 2, \ldots\ldots, k.$$

Note that the k selected vectors have been chosen from the set of m^i vectors

$$(-m+1, -m+1, \ldots\ldots, -m+1, I_1, I_2, \ldots\ldots, I_i)$$

where each I_j may take on the m values:

$$-m+1, -m+3, \ldots\ldots, m-1.$$

Hence the k selected vectors are distinct. (This result is needed only for the justification of an earlier comment).

System B, 1st cycle

At the start of the first cycle, System B has mk stored vectors which are set equal to the N+1 component vector

$$(-m+1, -m+1, \ldots\ldots, -m+1)$$

One of these vectors is assigned a zero cost and the others, infinite costs (see Section 3.02).

The estimated channel vectors for Systems A and B are:

$(y_1, y_2, \ldots\ldots, y_g)$ and

$(0, y_1, y_2, \ldots\ldots, y_g)$

respectively, therefore

$$\underline{Y}^B = (\underline{Y}^A, 0) \qquad (3.10)$$

where $\underline{Y}^B$ and $\underline{Y}^A$ are formed by reversing the channel vectors for System B and System A respectively. Therefore the received signal samples will be the same for Systems A and B, except that System B will receive an extra sample r_0 at the beginning. But, from equation 2.07,

$$r_o = \sum_{h=-g}^{0} s_h \, y_{o-h} + w_o$$

where $\{s_i\}$ is the sequence of symbols sent out by the transmitter, and $\{w_i\}$ is a sequence of noise samples. In defining System B, y_o was set equal to zero, therefore

$$r_o = \sum_{h=-g}^{-1} s_h \, y_{-h} + w_o$$

and it can be seen that r_o does not contain any information about the data sequence

$$s_0, s_1, s_2, \ldots\ldots$$

Hence the extra signal sample r_0, received by System B, does not represent the data sequence, and the received sequences for the two systems are effectively the same.

Following the algorithm for System B (see Section 3.02), the set of m^2k expanded vectors will contain m vectors of the form

$$(-m+1, -m+1, \ldots\ldots, -m+1, J_1)$$

in the first cycle. The costs for these m vectors are:

$$\{\underline{Y}^B . [-m+1, -m+1, \ldots.., -m+1, J_1]_{g+1} - r_o\}^2$$

where J_1 may take on the m values -m+1, -m+3, ..., m-1. The set of expanded vectors will also contain m^2k-m vectors with infinite costs.

Now

$$\underline{Y}^B . [-m+1, -m+1, \ldots.., -m+1, J_1]_{g+1}$$

is the scalar product of $\underline{Y}^B$ and the vector formed from the g+1 components of

$$[-m+1, -m+1, \ldots.., -m+1, J_1]$$

which are furthest to the right. Hence applying equation 3.10,

$$\underline{Y}^B . [-m+1, -m+1, \ldots.., -m+1, J_1]_{g+1} = \underline{Y}^A . [-m+1, -m+1, \ldots., -m+1]_g$$

Hence the m vectors of the form

$$(-m+1, -m+1, \ldots.., -m+1, J_1),$$

in the set of expanded vectors, have costs equal to c, where

$$c = \{\underline{Y}^A . [-m+1, -m+1, \ldots.., -m+1]_g - r_o\}^2$$

The mk vectors selected during the first cycle, are the ones with smallest costs. Hence this set of vectors will contain m vectors of the form

$$(-m+1, -m+1, \ldots.., -m+1, J_1)$$

with costs equal to c, and mk-m vectors with infinite costs.

System B, 2nd cycle

The expanded set of m^2k vectors will contain m^2 vectors of the form

$$(-m+1, -m+1, \ldots\ldots, -m+1, J_1\ J_2)$$

with costs equal to

$$c + \{\underline{Y}^B. [-m+1, -m+1, \ldots\ldots, -m+1, J_1, J_2]_{g+1} - r_1\}^2$$

(see Section 3.02). The expanded set will also contain $m^2k - m^2$ vectors with infinite costs. Now, applying equation 3.10, the costs for the first m^2 vectors may be written as

$$c + \{\underline{Y}^A. [-m+1, -m+1, \ldots\ldots, -m+1, J_1]_g - r_1\}^2$$

The set of mk selected vectors will contain these first m^2 vectors, plus $mk-m^2$ vectors with infinite costs.

System B, (i+1)st cycle

The set of m^2k expanded vectors will contain m^{i+1} vectors of the form

$$(-m+1, -m+1, \ldots\ldots, -m+1, J_1, J_2, \ldots\ldots, J_{i+1})$$

where each J_j may take on the m values

$$-m+1, -m+3, \ldots\ldots, m-1.$$

The costs for these m^{i+1} vectors are

$$c + \sum_{j=1}^{i} \{\underline{Y}^A . [-m+1, -m+1, \ldots\ldots, -m+1, J_1, J_2, \ldots\ldots, J_j]_g - r_j\}^2$$

The set of expanded vectors will also contain $m^2k - m^{i+1}$ vectors with infinite costs.

i has been defined to be the smallest integer such that $m^i \geq k$, so the number m^{i+1}, of vectors with finite costs, is greater than or equal to mk. Hence the mk vectors selected by decision rule 1 will all have finite costs, and will come from the set of vectors of the form

$$(-m+1, -m+1, \ldots\ldots, -m+1, J_1, J_2, \ldots\ldots, J_{i+1})$$

The values

$$[I_1(\ell), I_2(\ell), \ldots\ldots, I_i(\ell)\ ]$$

for $\ell = 1, 2, \ldots\ldots, k$

have been defined to be the values for

$$(I_1, I_2, \ldots\ldots, I_i)$$

which give the k smallest values for the cost function

$$\sum_{j=1}^{i} \{\underline{Y}^A . [-m+1, -m+1, \ldots\ldots, -m+1, I_1, I_2, \ldots\ldots, I_j]_g - r_j\}^2$$

(see the analysis of System A, cycle i).

Hence the mk vectors of the form

$$(-m+1, -m+1, \ldots\ldots, -m+1, J_1, J_2, \ldots\ldots, J_{i+1})$$

which give the mk smallest values for the cost function

$$c + \sum_{j=1}^{i} \{\underline{Y}^A . [-m+1, -m+1, \ldots\ldots, -m+1, J_1, J_2, \ldots\ldots, J_j]_g - r_j\}^2$$

are the vectors

$$[-m+1, -m+1, \ldots\ldots, -m+1, I_1(\ell), I_2(\ell), \ldots\ldots, I_i(\ell), J_{i+1}]$$

for $\ell = 1, 2, \ldots\ldots, k$

and $J_{i+1} = -m+1, -m+3, \ldots\ldots, m-1$.

These are the mk vectors selected by System B in the (i+1)st cycle.

Now consider cycle i of System A and cycle i+1 of System B. These are the cycles in which Systems A and B receive the signal sample r_i. Let the k vectors selected by System A, in cycle i, be denoted

$$\underline{R}(I)$$

for $I = 1, 2, \ldots\ldots, k$.

and let the corresponding costs be denoted u(I). Then the mk vectors retained by System B in the (i+1)st cycle are

$$[\underline{R}(I), J]$$

for $I = 1, 2, \ldots\ldots, k$

and $J = -m+1, -m+3, \ldots\ldots, m-1$.

The corresponding costs for these mk vectors are c + u(I). (They are independent of the value of J).

Now, applying lemma 3.01, it can be seen that this relationship between the vectors and costs of Systems A and B, will be maintained at the end of the next cycle of the algorithms. Clearly lemma 3.01 can be applied again and again, and this relationship will always be maintained. (It has been assumed here that the k costs u(I) are always distinct).

Of the N+1 component vectors $\underline{R}$(I) stored by System A during some cycle, the component furthest to the left of the vector with smallest cost, is taken as a detected element. Similarly, in the corresponding cycle of System B, the earliest element of the N+2 component vector [$\underline{R}$(I), J] with smallest cost, is taken as a detected element. But it can be seen from the above analysis, that the vectors $\underline{R}$(I) of System A, have the same costs as the vectors [$\underline{R}$(I), J] of System B. Therefore both Systems will produce the same detected data element. This is clearly true for all following cycles of the algorithms.

End of proof of theorem 3.02.

From theorem 3.02 it can be seen that, if the estimated channel vector

$$(0, y_1, y_2, \ldots\ldots, y_g)$$

is used with System 1, instead of the channel vector

$$(y_1, y_2, \ldots\ldots, y_g),$$

then m times as many vectors are needed to produce the same detected data sequence. Hence, if the number k of vectors stored by System 1, is a multiple of m, then the extra zero effectively reduces k by a factor of m. The number of components of the vectors is also effectively reduced by one.

The effect of an extra zero, at the start of the channel's sampled impulse response, will now be examined for System 2. Define System C to be System 2 with the estimated channel vector

$$(y_0, y_1, \ldots\ldots, y_g)$$

and mk N+1 component vectors stored at the start of each cycle. y_0 is defined equal to zero, as for System B.

Lemma 3.02

As in lemma 3.01, let the k vectors stored by System A at the start of some cycle, be $\underline{Q}(I)$, with a distinct set of costs u(I), for

$$I = 1, 2, \ldots\ldots, k.$$

Let the signal sample received by System A, in this cycle, be r. Suppose that the mk vectors stored by System C, in the cycle in which r is received, are

$$[\underline{Q}(I), J]$$

for $I = 1, 2, \ldots., k$

and $J = -m+1, -m+3, \ldots\ldots, m-1$.

Suppose also that the costs for these mk vectors are independent of J, and equal to c + u(I) for some constant c. Then this relationship, between the vectors and costs of Systems A and C, will be maintained at the start of the following cycle of the process.

Proof

The proof of this lemma is similar to that for lemma 3.01.

For system A, the set of mk expanded vectors is

$$[\underline{Q}(I), K]$$

for $I = 1, 2, \ldots\ldots, k$

and $K = -m+1, -m+3, \ldots\ldots, m-1$.

The costs for these vectors are given by

$$v(I, K) = u(I) + \{\underline{Y}^A . [\underline{Q}(I), K]_g - r\}^2 \tag{3.11}$$

where

$$\underline{Y}^A = (y_g, y_{g-1}, \ldots\ldots, y_1)$$

(see Section 3.02). Let the k pairs of values (I, K), which give the k smallest values of v(I, K), be denoted

$$[I(\ell), K(\ell)]$$

for $\ell = 1, 2, \ldots\ldots, k$.

Then the k vectors selected for System A are

$$[\underline{Q}(I(\ell)), K(\ell)]$$

for $\ell = 1, 2, \ldots\ldots, k$.

For System C, the expanded set of m^2k vectors is

$$[\underline{Q}(I), J, L]$$

for $I = 1, 2, \ldots\ldots, k$.

$J = -m+1, -m+3, \ldots\ldots, m-1$

and $L = -m+1, -m+3, \ldots\ldots, m-1$.

The costs for these vectors are given by

$$D(I, J, L) = c + u(I) + \{\underline{Y}^A . [\underline{Q}(I), J]_g - r\}^2$$

(see equation 3.08). From this equation and equation 3.11, it follows that

$$D(I, J, L) = c + v(I, J) \tag{3.12}$$

System C is a System 2 process, so it uses decision rule 2. Hence k vectors of the form

$$[\underline{Q}(I), J, L]$$

must be selected, for each possible value of L, i.e. the k values of (I, J), giving the smallest values of the cost function D(I, J, L), must be found for each value of L. However, from equation 3.12, it can be seen that the value of D(I, J, L) is independent of L. Hence, for any given value of L, the k pairs of values (I, J) which minimise D(I, J, L), are the ones which minimise v(I, J). These values have been denoted

$$[I(\ell), K(\ell)]$$

for $\ell = 1, 2, \ldots\ldots, k$.

Hence, for a given value of L, the k vectors selected by System C are

$$[\underline{Q}(I(\ell)), K(\ell), L]$$

for $\ell = 1, 2, \ldots\ldots, k$. The mk vectors selected by System C are

$$[\underline{Q}(I(\ell)), K(\ell), L]$$

for $\ell = 1, 2, \ldots\ldots, k$

and $L = -m+1, -m+3, \ldots\ldots, m-1$.

The costs for these vectors are independent of L and are equal to

$$c + v[I(\ell), K(\ell)]$$

These are the costs for the k vectors

$$[\underline{Q}(I(\ell)), K(\ell)]$$

selected by System A, except for the constant c. Hence the given relationship between the vectors and costs of Systems A and C, will be maintained at the start of the following cycle.

End of proof.

Theorem 3.03

Systems A and C will produce the same detected data sequence.

Proof

Details of the cycles of System A, up to the i th cycle, are given in the proof of theorem 3.02, where i is defined to be the least integer such that $m^i \geq k$.

System C, 1st cycle

Initially System C has mk stored vectors, each equal to the N+1 component vector

$$(-m+1, -m+1, \ldots\ldots, -m+1)$$

One of these vectors is assigned a zero cost and the others, infinite costs. (See Section 3.02). It will be seen that System C produces the same vectors and costs, in every cycle, as System B.

Let $\underline{Y}^C = (y_g, y_{g-1}, \ldots.., y_0)$

where $y_0 = 0$, so that $\underline{Y}^C$ is the reverse of the estimated channel vector for System C. Then

$$\underline{Y}^C = (\underline{Y}^A, 0) \tag{3.13}$$

and $\underline{Y}^C$ is the same as $\underline{Y}^B$. As for System B, the first signal sample r_0, received by System C, is independent of the data sequence.

The expanded set of m^2k vectors will contain m vectors with finite costs, of the form

$$(-m+1, -m+1, \ldots.., -m+1, J_1)$$

where J_1 can take on the values

$$-m+1, -m+3, \ldots.., m-1.$$

The costs for these m vectors are

$$\{\underline{Y}^C. [-m+1, -m+1, \ldots.., -m+1, J_1]_{g+1} - r_0\}^2$$

(See Section 3.02). The set of expanded vectors will also contain $m^2k - m$ vectors with infinite costs.

Using equation 3.13, it can be seen that

$$\underline{Y}^C. [-m+1, -m+1, \ldots., -m+1, J_1]_{g+1} = \underline{Y}^A. [-m+1, -m+1, \ldots., -m+1]_g$$

Hence the first m vectors in the set of expanded vectors, have costs

$$\{\underline{Y}^A . [-m+1, -m+1, \ldots., -m+1]_g - r_0\}^2 = c, \text{ say.}$$

Now, according to decision rule 2, the k vectors of the form

$$(-m+1, -m+1, \ldots\ldots, -m+1, J_1)$$

with smallest costs, must be seleted for each possible value of J_1. In the expanded set, there is one vector with cost c, and k-1 vectors with infinite costs, for each value of J_1. Hence the m vectors with cost c must be among those selected. The set of mk selected vectors will therefore contain the m vectors

$$(-m+1, -m+1, \ldots\ldots, -m+1, J_1)$$

for $J_1 = -m+1, -m+3, \ldots\ldots, m-1,$

with costs equal to c. mk-m vectors with infinite costs, will also be selected.

System C, 2nd cycle

As for System B, the set of m^2k expanded vectors will contain m^2 vectors of the form

$$(-m+1, -m+1, \ldots\ldots, -m+1, J_1, J_2)$$

with costs

$$c + \{\underline{Y}^A. [-m+1, -m+1, \ldots\ldots, -m+1, J_1]_g - r_1\}^2$$

The expanded set will also contain m^2k-m^2 vectors with infinite costs.

For each value of J_2, the k vectors of the form

$$(-m+1, -m+1, \ldots\ldots, -m+1, J_1, J_2)$$

with smallest costs, will be selected by System C, (i.e. by decision rule 2). There are m such vectors with finite costs, for each of the m values of J_2, so all of the vectors with finite costs will be selected (assuming that $k \geq m$). Hence the set of mk selected vectors will contain m^2 vectors of the form

$$(-m+1, -m+1, \ldots\ldots, -m+1, J_1, J_2)$$

with costs

$$c + \{\underline{Y}^A . [-m+1, -m+1, \ldots\ldots, -m+1, J_1]_g - r_1\}^2$$

This set will also contain $mk - m^2$ vectors with infinite costs.

System C, (i+1)st cycle

The set of m^2k expanded vectors will contain m^{i+1} vectors of the form

$$(-m+1, -m+1, \ldots\ldots ,-m+1, J_1, J_2, \ldots\ldots, J_{i+1})$$

with costs equal to

$$c + \sum_{j=1}^{i} \{\underline{Y}^A . [-m+1, -m+1, \ldots\ldots, -m+1, J_1, J_2, \ldots\ldots, J_j]_g - r_j\}^2.$$

$m^2k - m^{i+1}$ vectors with infinite costs will also be included in the expanded set. i has been defined to be the smallest integer such that $m^i \geq k$, so there are mk or more vectors in the expanded set, with finite costs. In fact, for each of the m values of J_{i+1}, there are k or more vectors of the form

$$[-m+1, -m+1, \ldots\ldots, -m+1, J_1, J_2, \ldots\ldots, J_{i+1}]$$

with finite costs. The k vectors with smallest costs must now be selected, for each value of J_{i+1}. Hence, for each value of J_{i+1}, the k selected vectors are given by the k values of

$$(J_1, J_2, \ldots\ldots, J_i)$$

which minimise the cost function

$$c + \sum_{j=1}^{i} \{\underline{Y}^A. [-m+1, -m+1, \ldots., -m+1, J_1, J_2, \ldots., J_j]_g - r_j\}^2$$

But these values have been denoted

$$[I_1(\ell), I_2(\ell), \ldots., I_i(\ell)]$$

for $\ell = 1, 2, \ldots., k$. (See the analysis of System A, cycle i).

Hence the mk vectors selected by System C, in this cycle are

$$[-m+1, -m+1, \ldots., -m+1, I_1(\ell), I_2(\ell), \ldots., I_i(\ell), J_{i+1}]$$

for $\ell = 1, 2, \ldots., k$.

and $J_{i+1} = -m+1, -m+3, \ldots\ldots, m-1$.

It can be seen that the vectors and costs stored at the end of the (i+1)st cycle, are the same for both System B and System C. Hence the argument used in theorem 3.02 also applies for this case with System C, except that lemma 3.02 must now be used in place of lemma 3.01. Systems A and C therefore produce the same detected data sequences.

End of proof of Theorem 3.03.

Now consider a situation with System 1 using the correct estimate

$$(y_1, y_2, \ldots\ldots, y_g)$$

of the channel's sampled impulse response. Then, from theorem 3.03, it can be seen that System 2 requires m times as many stored vectors as System 1, to produce the same detected data sequence, if it uses the incorrect estimate:

$$(0, y_1, y_2, \ldots\ldots, y_g)$$

of the channel vector.

The simulation tests described in Chapter 4, show that the performances of Systems 1 and 2 are usually about the same, for a given number of stored vectors. Hence, where the number k of vectors stored by System 2, is a multiple of m^2, inserting an extra zero at the start of the channel vector, effectively reduces the number of stored vectors by a factor of m. Note that System C was defined to be operating with N+1 components in each vector, whereas the vectors of System A had N components. Hence the addition of the extra zero, also effectively reduces the number of these components by one.

The effect of the extra zero, at the start of the channel vector, will now be investigated for System 4. Define System D to be System 4, with the estimated channel vector

$$(y_1, y_2, \ldots\ldots, y_g),$$

and k N component vectors stored at the start of each cycle.

Also define System E to be System 4, with the estimated channel vector

$$(y_0, y_1, \ldots\ldots, y_g)$$

and mk N+1 component vectors stored at the start of each cycle, where $y_0 = 0$.

Lemma 3.03

Consider some cycle of the detection process, in which the signal sample r is received by System D. Let the k vectors stored at the start of this cycle be denoted $\underline{Q}(I)$ with a set of distinct costs u(I), for I = 1, 2, ..., k. Suppose that the mk vectors of System E, at the start of the cycle in which the sample r is received, are

$$[\underline{Q}(I), J]$$

for $I = 1, 2, \ldots\ldots, k$.

and $J = -m+1, -m+3, \ldots\ldots, m-1$

Suppose also that the cost for each vector

$$[\underline{Q}(I), J]$$

is independent of J, and equal to

$$c + u(I)$$

for some constant c. Then this relationship between the vectors and costs of Systems D and E, will be maintained at the start of the next cycle of the algorithm.

Proof

System D is a particular case of System 4, so the set of k vectors

$$\underline{Q}(1),\ \underline{Q}(2),\ \ldots\ldots,\ \underline{Q}(k)$$

contain all possible combinations of the latest ℓ components, where ℓ is given by

$$k = m^{\ell}$$

Hence these k vectors may be divided into m sets, of the form $\underline{Q}^{I}(K)$, where I is the value of the component which is ℓth from the right. Then each value of K corresponds to a particular combination of the latest ℓ-1 components of the vectors. I may take on the m values -m+1, -m+3, ..., m-1,
and K may take the values,

$$1,\ 2,\ \ldots\ldots,\ k/m.$$

For System D, the set of mk expanded vectors are of the form

$$[\underline{Q}^{I}(K),\ L]$$

for $K = 1, 2, \ldots\ldots, k/m$

$I = -m+1, -m+3, \ldots\ldots, m-1$

and $L = -m+1, -m+3, \ldots\ldots, m-1.$

The costs for these vectors are given by

$$v(I,\ K,\ L) = u^{I}(K) + \{\underline{Y}^{D}.\ [\underline{Q}^{I}(K),\ L]_{g} - r\}^{2} \qquad (3.14)$$

(see Section 3.02), where $u^{I}(K)$ is the cost for the vector $\underline{Q}^{I}(K)$ and

$$\underline{Y}^D = (y_g, y_{g-1}, \ldots\ldots, y_1).$$

The value of K dictates a particular combination of the latest ℓ-1 components of $\underline{Q}^I(K)$. Hence the values of K and L dictate a particular combination of the latest ℓ elements of the vector

$$[\underline{Q}^I(K), L].$$

According to the decision rule for System D (i.e. decision rule 4), the vector $[\underline{Q}^I(K), L]$ with smallest cost, must be selected for each possible combination of the latest ℓ components. Hence, for each value of (K, L) one value of I must be chosen. Let this value of I be denoted I(K, L). Then, for given values of K and L, the value of I which minimises the cost function v(I, K, L) is I(K, L). Hence the k vectors selected by System D are

$$[\underline{Q}^P\ (K), L] \quad \text{with costs:} \quad v[P, K, L]$$

for $K = 1, 2, \ldots\ldots, k/m$

and $L = -m+1, -m+3, \ldots\ldots, m-1$

with $P = I(K, L)$.

For System E, the set of m^2k expanded vectors is

$$[\underline{Q}^I(K), J, L]$$

for $K = 1, 2, \ldots\ldots, k/m$

$J = -m+1, -m+3, \ldots\ldots, m-1$

and $L = -m+1, -m+3, \ldots\ldots, m-1$.

The costs for these vectors are given by

$$D(I, K, J, L) = c + u^I(K) + \{\underline{Y}^E.\ [\underline{Q}^I(K), J, L]_{g+1} - r\}^2 \qquad (3.15)$$

(See Section 3.02), where

$$\underline{Y}^E = (y_g, y_{g-1}, \ldots\ldots, y_1, 0).$$

Clearly, by definition,

$$\underline{Y}^E = (\underline{Y}^D, 0)$$

therefore

$$\underline{Y}^E. [\underline{Q}^I(K), J, L]_{g+1} = \underline{Y}^D. [\underline{Q}^I(K), J]_g$$

Therefore, equation 3.15 becomes

$$D(I, K, J, L) = c + u^I(K) + \{\underline{Y}^D. [\underline{Q}^I(K), J]_g - r\}^2$$

Hence, from this equation and equation 3.14

$$D(I, K, J, L) = c + v(I, K, J) \qquad (3.16)$$

The value of K dictates a particular combination of the latest ℓ-1 elements of $\underline{Q}^I(K)$. Hence the values of K, J and L dictate a particular combination of the latest ℓ+1 elements of the vector

$$[\underline{Q}^I(K), J, L]$$

Now, according to the decision rule for System E, the vector with smallest cost must be selected, for each possible combination of the latest ℓ+1 components. Hence, for each value of (K, J, L), the value of I which minimises the cost function D(I, K, J, L), must be found. But, from equation 3.16, the value of I which minimises D(I, K, J, L) for given values of J, K and L, is the value which minimises v(I, K, J). This value of I has been denoted I(K, J). Hence the mk selected vectors for System E are

$$[\underline{Q}^P(K), J, L]$$

for $K = 1, 2, \ldots\ldots, k/m$

$J = -m+1, -m+3, \ldots\ldots, m-1$

and $L = -m+1, -m+3, \ldots\ldots, m-1$

with $P = I(K, J)$.

The costs for these vectors are

$$c + v[P, K, J]$$

where

$$v[P, K, J]$$

is the cost for the vector

$$[\underline{Q}^P(K), J]$$

selected by System D. Hence the relationship between the vectors and costs of Systems D and E, that existed at the start of the cycle, is maintained at the start of the next cycle of the process.

End of proof of lemma 3.03.

Theorem 3.04

Systems D and E will produce the same detected data sequence.

Proof:

System D, 1st cycle

Initially, System D has k stored vectors, each equal to the N component vector

$$(-m+1, -m+1, \ldots\ldots, -m+1)$$

One of the vectors is assigned a cost of zero and the other vectors are given infinite costs. The set of mk expanded vectors contains m vectors of the form.

$$(-m+1, -m+1, \ldots\ldots, -m+1, I_1)$$

with costs

$$\{\underline{Y}^D . [-m+1, -m+1, \ldots\ldots, -m+1, I_1]_g - r_1\}^2$$

where r_1 is the first signal sample received by System D, and I_1 may take on the m values

$$-m+1, -m+3, \ldots\ldots, m-1$$

(see Section 3.02). The set of expanded vectors also contains mk-m vectors with infinite costs. System D uses decision rule 4, so the vector with lowest cost must be selected, for each possible value of I_1. The set of k selected vectors is then completed with an arbitrary selection from the remaining mk-m vectors. The k selected vectors will therefore contain m vectors of the form

$$(-m+1, -m+1, \ldots\ldots, -m+1, I_1)$$

with costs

$$\{\underline{Y}^D.\ [-m+1,\ -m+1,\ \ldots\ldots,\ -m+1,\ I_1]_g\ -\ r_1\}^2.$$

k-m vectors with infinite costs are also contained in the set of selected vectors.

System D, 2nd cycle

The set of mk extended vectors will contain m^2 vectors of the form

$$(-m+1,\ -m+1,\ \ldots\ldots,\ -m+1,\ I_1,\ I_2)$$

with costs

$$\sum_{j=1}^{2}\ \{\underline{Y}^D.\ [-m+1,\ -m+1,\ \ldots\ldots,\ -m+1,\ I_1,\ I_2,\ \ldots\ldots,\ I_j]_g\ -\ r_j\}^2$$

The set of extended vectors will also contain mk-m^2 vectors with infinite costs. Now, according to decision rule 4, the vector with lowest cost must be selected for each possible combination of the latest two components. (For components other than the latest two, only one value is available in all of the mk vectors). The set of k selected vectors will therefore contain m^2 vectors of the form

$$(-m+1,\ -m+1,\ \ldots\ldots,\ -m+1,\ I_1,\ I_2)$$

with costs

$$\sum_{j=1}^{2}\ \{\underline{Y}^D.\ [-m+1,\ -m+1,\ \ldots.,\ -m+1,\ I_1,\ I_2,\ \ldots.,\ I_j]_g\ -\ r_j\}^2$$

This set will also contain k-m² vectors with infinite costs (assuming that $k \geq m^2$).

System D, ℓth cycle

The set of mk extended vectors will contain m^ℓ vectors of the form

$$(-m+1,\ -m+1,\ \ldots.,\ -m+1,\ I_1,\ I_2,\ \ldots..,\ I_\ell)$$

with costs

$$\sum_{j=1}^{\ell} \{\underline{Y}^D.\ [-m+1,\ -m+1,\ \ldots.,\ -m+1,\ I_1,\ I_2,\ \ldots.,\ I_j]_g - r_j\}^2$$

The expanded set will also contain $mk - m^\ell$ vectors with infinite costs. ℓ has been defined to be the integer such that $k = m^\ell$, so this set of mk vectors contains k vectors with finite costs. The vector with smallest cost must now be chosen, for each possible combination of the latest ℓ components. Hence all of the vectors with finite costs will be selected. The k selected vectors will be of the form

$$(-m+1,\ -m+1,\ \ldots.,\ -m+1,\ I_1,\ I_2,\ \ldots..,\ I_\ell)$$

with costs

$$\sum_{j=1}^{\ell} \{\underline{Y}^D.\ [-m+1,\ -m+1,\ \ldots.,\ -m+1,\ I_1,\ I_2,\ \ldots.,\ I_j]_g - r_j\}^2$$

System E, 1st cycle

Initially, System E has mk stored vectors, each equal to the N+1 component vector

$$(-m+1,\ -m+1,\ \ldots.,\ -m+1)$$

One of the vectors is assigned a zero cost and the others, infinite costs.

The set of m^2k extended vectors will contain m vectors of the form

$$(-m+1,\ -m+1,\ \ldots..,\ -m+1,\ J_1)$$

with costs

$$\{\underline{Y}^E.\ [-m+1,\ -m+1,\ \ldots..,\ -m+1,\ J_1]_{g+1} - r_0\}^2$$

where r_0 is the first signal sample received by System E. J_1 may take on the m values

$$-m+1,\ -m+3,\ \ldots..,\ m-1$$

and $\underline{Y}^E = (y_g,\ y_{g-1},\ \ldots..,\ y_1,\ 0)$

From the definitions of $\underline{Y}^D$ and $\underline{Y}^E$, it can be seen that

$$\underline{Y}^E = (\underline{Y}^D,\ 0)$$

Hence

$$\underline{Y}^E.\ [-m+1,\ -m+1,\ \ldots.,\ -m+1,\ J_1]_{g+1} = \underline{Y}^D.\ [-m+1,\ -m+1,\ \ldots.,\ -m+1]_g$$

Hence the set of extended vectors contains m vectors of the form

$$(-m+1,\ -m+1,\ \ldots\ldots,\ -m+1,\ J_1)$$

with costs c, where

$$c = \{\underline{Y}^D.\ [-m+1,\ -m+1,\ \ldots\ldots,\ -m+1]_g - r_o\}^2$$

(These costs are independent of J_1). The set of extended vectors also contains m^2k-m vectors with infinite costs.

According to decision rûle 4, the vector with smallest cost must be selected, for each of the m possible values of J_1. The set of selected vectors is then made up by choosing mk-m vectors arbitrarily from those remaining. Hence the set of k selected vectors contains mk-m vectors with infinite costs and m vectors of the form

$$(-m+1,\ -m+1,\ \ldots\ldots,\ -m+1,\ J_1)$$

with costs equal to c.

System E, 2nd cycle

The set of m^2k extended vectors will contain mk-m^2 vectors with infinite costs, and m^2 vectors of the form

$$(-m+1,\ -m+1,\ \ldots\ldots,\ -m+1,\ J_1,\ J_2)$$

with costs

$$c + \{\underline{Y}^D.\ [-m+1,\ -m+1,\ \ldots\ldots,\ -m+1,\ J_1]_g - r_1\}^2$$

The vector with smallest cost must be selected for each possible combination of the latest two elements, thus giving m^2 selected vectors. The set of selected vectors is then completed with an arbitrary choice from the remaining m^2k-m^2 vectors. Hence the set of mk selected vectors will contain $mk-m^2$ vectors with infinite costs, and m^2 vectors of the form

$$(-m+1,\ -m+1,\ \ldots\ldots,\ -m+1,\ J_1,\ J_2)$$

with costs equal to

$$c + \{\underline{Y}^D.\ [-m+1,\ -m+1,\ \ldots.,\ -m+1,\ J_1]_g - r_1\}^2$$

System E, (ℓ+1)st cycle

The set of m^2k extended vectors will contain $m^2k-m^{\ell+1}$ vectors with infinite costs, and $m^{\ell+1}$ vectors of the form

$$(-m+1,\ -m+1,\ \ldots\ldots,\ -m+1,\ J_1,\ J_2,\ \ldots\ldots,\ J_{\ell+1})$$

with costs equal to

$$c + \sum_{j=1}^{\ell} \{\underline{Y}^D.\ [-m+1,\ -m+1,\ \ldots\ldots,\ -m+1,\ J_1,\ J_2,\ \ldots\ldots,\ J_j]_g - r_j\}^2.$$

ℓ is defined to be the integer such that $k = m^\ell$, so there are now mk vectors with finite costs. The vector with smallest cost must be selected for each combination of the latest $\ell+1$ components, so the mk selected vectors are of the form

$$(-m+1,\ -m+1,\ \ldots\ldots,\ -m+1,\ J_1,\ J_2,\ \ldots\ldots,\ J_{\ell+1})$$

with costs equal to

$$c + \sum_{j=1}^{\ell} \{\underline{Y}^D.\ [-m+1,\ -m+1,\ \ldots\ldots,\ -m+1,\ J_1,\ J_2,\ \ldots.,\ J_j]_g - r_j\}^2$$

Now consider cycle ℓ of System D and cycle $\ell+1$ of System E. Denote the k vectors selected by System D, in cycle ℓ, by $\underline{R}(I)$ and their costs by $u(I)$, for $I = 1, 2, \ldots.., k$. (These k costs are assumed to be distinct). Then, from the above analysis it can be seen that the mk vectors selected by System E, in its $\ell+1$st cycle are

$[\underline{R}(I), J]$ with costs

$c + u(I)$

for $I = 1, 2, \ldots.., k$

and $J = -m+1, -m+3, \ldots.., m-1$.

From lemma 3.03, this relationship between the vectors and costs of Systems D and E, will be maintained at the end of the following cycle of the process. Clearly the lemma can be applied again and again, so this relationship will be maintained during all future cycles.

Of the kN+1 component vectors of System D, the one with smallest cost gives the detected data element in each cycle. The element detected is the component furthest to the left, of the vector $\underline{R}(I)$ with smallest cost. With System E, the component furthest to the left, of the N+2 component vector $(\underline{R}(I), J)$ with smallest cost, is detected. It can be seen from the above analysis, that the costs for the vectors $\underline{R}(I)$ of System D, are the same as the costs for the vectors $[\underline{R}(I), J]$ of System E. Therefore both Systems will produce the same detected data element. As the vectors and costs of the two Systems remain linked, in the manner indicated above, it is clear that all detected elements will be identical for the Systems.

End of proof of theorem 3.04.

From theorem 3.04, it can be seen that, if the estimated channel vector

$$(0, y_1, y_2, \ldots\ldots, y_g)$$

is used instead of

$$(y_1, y_2, \ldots\ldots, y_g),$$

then m times as many stored vectors are needed by System 4, if the same detected data sequence is to be produced. Hence inserting a zero at the start of the channel vector, effectively reduces the number of stored vectors by a factor of m, (where m is the number of signal levels). The number of components of the stored vectors is also effectively reduced, by one. By means of theorems 3.02 and 3.03, it has been shown that this result also holds for System 1, and is approximately true for System 2. However no such result appears to be available for System 3. The simulation results of Chapter 4 show, in fact, that System 3 is affected much more severely than the other systems, by the presence of an extra zero at the start of the channel vector.

3.11 The Effect of an Extra Small Component at the Start of the Channel's Sampled Impulse Response

It has been shown in Section 3.10, that adding a zero at the start of the channel vector, effectively reduces the number of stored vectors by a factor of m, for Systems 1, 2 and 4. It seems reasonable that this result should also hold approximately, if a small value is added to the channel vector instead of a zero.

This is shown to be the case, at least for some channels, by means of the simulation results presented in Chapter 4.

If the first component of the channel's sampled impulse response is very small, it is probably best for the detector to ignore this component completely. The estimated sampled impulse response would then be

$$(y_1, y_2, \ldots\ldots, y_g) \quad \text{instead of} \quad (y_0, y_1, \ldots\ldots, y_g)$$

An alternative method for improving the performance of the detection processes, when y_0 is small, will now be considered. Let the z transform of the channel's sampled impulse response be Y(z). (The z transform was defined in Section 1.11). Let the roots of Y(z) with modulus greater than one be denoted

$$\alpha_1, \alpha_2, \ldots\ldots, \alpha_p,$$

and let the roots with modulus less than or equal to one be

$$\beta_1, \beta_2, \ldots\ldots, \beta_q.$$

Then

$$Y(z) = c(z^{-1}- \alpha_1{}^{-1})(z^{-1}-\alpha_2{}^{-1})\ldots(z^{-1}-\alpha_p{}^{-1})(z^{-1}-\beta_1{}^{-1})(z^{-1}-\beta_2{}^{-1})\ldots(z^{-1}-\beta_q{}^{-1}$$

for some constant c. Note that Y(z) may be written as a power series in z^{-1}. The constant term of the power series is the first component of the channel's sampled impulse response. Hence

$$y_0 = c\,\alpha_1{}^{-1}\,\alpha_2{}^{-1} \ldots\ldots \alpha_p{}^{-1}\,\beta_1{}^{-1}\,\beta_2{}^{-1} \ldots\ldots \beta_q{}^{-1}\,(-1)^{p+q}$$

where all of the terms α_i have modulus greater than one. Clearly, if the channel vector could be transformed so that the α_i terms are replaced by their reciprocals, then y_0 would be increased in size. Hence consider the linear filter whose z transform is given by

$$Y^*(z) = \frac{(\alpha_1{}^{-1}z^{-1}-1)(\alpha_2{}^{-1}z^{-1}-1) \;.....\; (\alpha_p{}^{-1}z^{-1}-1)}{(z^{-1}-\alpha_1{}^{-1})(z^{-1}-\alpha_2{}^{-1}) \;.....\; (z^{-1}-\alpha_p{}^{-1})}$$

If this filter is placed between the sampled received signal and the detector, the sampled impulse response estimated by the receiver will have z transform given by

$$Y(z)\; Y^*(z) = c(\alpha_1{}^{-1}z^{-1}-1)(\alpha_2{}^{-1}z^{-1}-1)\ldots(\alpha_p{}^{-1}z^{-1}-1)(z^{-1}-\beta_1{}^{-1})$$

$$\times\; (z^{-1}-\beta_2{}^{-1})\ldots(z^{-1}-\beta_q{}^{-1})$$

Now the constant term of the power series is

$$c\;\beta_1{}^{-1}\;\beta_2{}^{-1}\;.....\;\beta_q{}^{-1}$$

which may be considerably larger than for the case without the linear filter. Hence the linear filter with z transform Y*(z) may be used to effectively alter the components of the channel vector, in such a way that the first component is increased in size. This may be expected to improve the performances of Systems 1-4.

Note that it may be shown, that the type of linear filter described above introduces only pure phase distortion in the received signal [6,14]. For a case where the channel characteristics vary with time, it may be of advantage to use a linear filter of this type, which adapts to the changes in the channel. Then the first component of the channel vector, as seen by the detector, will always be reasonably large. Some simulation results are presented in Chapter 4, which demonstrate the improvement in performance offered by such a filter.

3.12 Probability of Error with Systems 1-4 when Used with the Ideal Channel

Each of the Systems 1-4, start off with a number k of N component vectors given by

$$\underline{Q}_{-1}(I) = (-m+1, -m+1, \ldots\ldots, -m+1)$$

The corresponding costs are given by

$$u_{-1}(I) = \begin{cases} 0 & \text{for } I = 1 \\ \infty & \text{for } I = 2, 3, \ldots\ldots, k \end{cases}$$

These vectors are extended to the mk N+1 component vectors

$$\underline{T}_0(I, x_0) = [\underline{Q}_{-1}(I), x_0]$$

$$= [-m+1, -m+1, \ldots\ldots, -m+1, x_0]$$

where x_0 may take on the values

$-m+1, -m+3, \ldots\ldots, m-1.$

The costs for these extended vectors are given by

$$v_0(I, x_0) = u_{-1}(I) + \{\underline{Y}.\ [\underline{T}_0(I, x_0)]_{g+1} - r_0\}^2$$

(see Section 3.02), where r_o is the first received signal sample, and $\underline{Y}$ is the vector formed by reversing the channel vector. Here, the channel vector being considered has just one component equal to unity, (this being the definition of the ideal channel), so

$$\underline{Y} = 1 \qquad \text{and} \qquad g = 0$$

Hence

$$v_o\ (I,\ x_o) = u_{-1}(I) + (x_o - r_o)^2.$$

Let s_o' be the data element value which is closest to r_o. Then the vector, from the set of mk extended vectors, with smallest cost is $\underline{T}_o(1,\ s_o')$, or

$$(-m+1,\ -m+1,\ \ldots\ldots,\ -m+1,\ s_o')$$

The cost for this vector is $(s_o' - r_o)^2$.

Now, according to the appropriate decision rule (either rule 1, 2, 3 or 4), k of the extended vectors will be selected and retained for use in the following cycle of the algorithm. All of the decision rules are designed in such a way, that the vector with lowest cost will be among those selected. Hence $\underline{T}_o(1,\ s_o')$ will be retained for the coming cycle.

In the second cycle, the mk extended vectors will be of the form

$$\underline{T}_1(I,\ x_1) = (-m+1,\ -m+1,\ \ldots\ldots,\ -m+1,\ x_o,\ x_1).$$

m of these vectors will have stemmed from the N component vector

$$(-m+1,\ -m+1,\ \ldots\ldots,\ -m+1,\ s_o')$$

with cost equal to $(s_o' - r_o)^2$. The costs for these m vectors are given by

$$v_1(1,\ x_1) = (s_o' - r_o)^2 + \{\underline{Y}.[-m+1,\ -m+1,\ \ldots,-m+1,\ s_o',x_1]_{g+1} - r_1\}^2$$

$$= (s_o' - r_o)^2 + (x_1 - r_1)^2$$

as $\underline{Y}$ is the scalar, 'one', and g = 0.

Now let s_1' be the data element value which is closest to r_1. Then the vector from the set $\{\underline{T}_1(I,\ x_1)\}$ with smallest cost is given by

$$\underline{T}_1(1,\ s_1') = (-m+1,\ -m+1,\ \ldots\ldots,\ -m+1,\ s_o',\ s_1')$$

and its cost is given by

$$\underline{v}_1(1,\ s_1') = (s_o' - r_o)^2 + (s_1' - r_1)^2$$

Similarly, in the third cycle of the algorithm, the extended vector with smallest cost is the N+1 component vector

$$(-m+1,\ -m+1,\ \ldots\ldots,\ -m+1,\ s_o',\ s_1',\ s_2')$$

where s_2' is the data element value which is closest to the third received sample r_2.

Now consider the (k+1)st cycle of the process, in which the signal sample r_k is received. Let the extended vector with smallest cost be denoted

$$(s'_{k-N} , s'_{k-N+1}, \ldots\ldots, s'_k)$$

Then it follows from the above analysis, that s_i' is the data element value which is closest to the received signal sample r_i, for

$$i = k-N, k-N+1, \ldots\ldots, k.$$

Each of the Systems 1-4 are designed in such a way that the data elements $\{s_i\}$ are detected as the component furthest to the left, of the vector with smallest cost, in each cycle. Hence each data element s_i, is detected as the data element value which is closest to the received signal sample r_i. Therefore, with the ideal channel, Systems 1-4 produce the same detected data sequence as the non linear equalizer described in Section 1.13. (See Section 1.14). This of course implies that the probability of error, in the detected sequence, is the same as that for the non linear equalizer, and is given by

$$P_e = \text{Prob } (w_i > 1)$$

for the case of binary signals

and

$$P_e = 1.5 \text{ Prob } (w_i > 1)$$

for quaternary signals. (w_i is a normally distributed random variable, with zero mean and given variance). It should be noted that these results are independent of the number k, of vectors stored by the detection processes.

CHAPTER 4

4.01 The Value of Computer Simulation Testing

When detection processes such as Systems 1-4 have been designed, perhaps the most obvious method for assessing their performances, would involve constructing a piece of hardware (an electronic circuit) which carries out the required operations. The detectors could then be used as part of a data transmission system, and the number of errors occurring may be measured. However this approach to the evaluation of the detection processes, does have some disadvantages. After the various pieces of hardware have been constructed, it may be desirable to make some modifications to the processes. Extensive and time consuming alterations may then be required, even for apparently small changes in the algorithms. The construction of the necessary electronic circuits itself, may also be a difficult and expensive task.

Another approach that should be considered, for evaluating the detectors, is one of thorough mathematical analysis. It would be very useful if an expression for the proportion of errors expected, in the detected data sequence, could be derived for Systems 1-4. Such a derivation would however appear to be a difficult task, due to the number and type of operations required by the algorithms. Forney [35] has obtained a bound for the probability of error in the detected data sequence, for the Viterbi Algorithm. It may be possible to apply a similar analysis to Systems 1-4.

The performances of the various detection processes were, in fact, evaluated by means of computer simulation tests. i.e. a program was written for a modern high speed digital computer, which performs the necessary operations on any given sequence of received signal samples. This method has the advantage, that the cost of the materials required to produce the program is negligible. Also, quite fundamental modifications can sometimes be made to the algorithms, by just retyping a few instructions.

The main disadvantage of computer simulation testing, seems to be that a very large amount of computing time is required, to obtain some types of performance figures. Even on a reasonably fast digital computer such as an ICL 1904A, some hundreds of hours of program run time would be needed to produce the data supplied in this chapter. Many of the longer program runs were carried out on a CDC 7600 computer, situated at Manchester University. The shorter runs were on the ICL 1904A at Loughborough University. The programming language used on the 1904A was 1900 Fortran, which is very similar to Fortran IV. Apart from a few statements having to be altered, these programs were also suitable for use on the CDC 7600.

One point to bear in mind with computer simulation testing, is that most computers can store numbers and perform calculations, to a high degree of accuracy. However, if a piece of hardware was constructed to implement the detection processes, a relatively crude calculation facility would probably be employed, to reduce costs. Hence the simulated detection processes, may be expected to perform slightly better than those implemented in practice.

4.02 Simulation of a Data Transmission System

The different detection processes were tested for the cases of a binary data signal (m=2), anda quaternary signal (m=4). The model of the data transmission system used, is described in Section 1.02. The elements s_i, of the data sequence, may take on the m values

$$-m+1,\ -m+3,\ \ldots\ldots,\ m-1$$

The different possible values of each s_i are assumed to be statistically independent and equally likely, so a standard subroutine was used to produce the data element values in a random manner. The NAG (Numerical Algorithms Group) subroutine, "G05AAF", was used to provide a simulated random number from a uniform (0, 1) distribution. This is a distribution with a probability density function f(x) such that

$$f(x) = \begin{cases} 1 & \text{for } 0 \le x \le 1 \\ 0 & \text{otherwise} \end{cases}$$

Let X be a sample from the simulated uniform distribution. Then, for a binary signal, s_0 is defined by

$$s_0 = \begin{cases} -1 & \text{if } 0 \le X < 0.5 \\ 1 & \text{if } 0.5 \le X \le 1 \end{cases}$$

For a quaternary signal,

$$s_0 = \begin{cases} -3 & \text{if } 0 \le X < 0.25 \\ -1 & \text{if } 0.25 \le X < 0.5 \\ 1 & \text{if } 0.5 \le X < 0.75 \\ 3 & \text{if } 0.75 \le X \le 1 \end{cases}$$

A new sample may then be called from the NAG subroutine, to define each of the other data elements.

From the data sequence $\{s_i\}$, a received signal sequence $\{r_i\}$ must be generated, such that

$$r_i = y_0 s_i + y_1 s_{i-1} + \ldots.. + y_g s_{i-g} + w_i$$

where $(y_0, y_1, \ldots.., y_g)$ is the sampled impulse response of the channel, and $\{w_i\}$ is a sequence of simulated random numbers representing noise. The samples w_i are assumed to be taken from a normal distribution, with zero mean and some fixed variance σ^2. Each w_i was provided by the standard NAG subroutine G05AEF, in which any desired mean, and standard deviation σ, may be specified. One subroutine of the computer programs was devoted to generating the sequence $\{r_i\}$, of received signal samples. This subroutine then represents a transmitter and a baseband channel. The remainder of the programs contained the operations necessary for implementing the detection processes.

4.03 Method of Comparison of the Detection Processes

Consider the model of a data transmission system, given in Section 1.02. With this model, errors will occur in the detected data sequence, if the average power level of the additive white Gaussian noise is sufficiently high. The proportion of errors occurring in the detected sequence, is a random variable whose expected value increases with the noise power. Note that it is fairly straightforward to determine the proportion of errors occurring in a simulation test, by simply comparing the generated data sequence with the detected sequence.

The criterion under which the different detection processes were compared, was that of their tolerance to additive white Gaussian noise. The tolerance to noise is just a measure of the noise standard deviation σ, which gives rise to some given expected error rate in the detected data sequence. Many of the tests were performed at an error rate of 0.004, i.e. on a long term average, there were 4 detected elements in error, out of every 1000. Suppose that, with one detection process, a larger noise level is required to produce an error rate whose expected value is 0.004, than with a second detector. Then the former detection process is said to have the greatest tolerance to additive white Gaussian noise, for the given conditions, and at an error rate of 0.004.

As an alternative, the various detectors could have been compared, by means of simulation tests in which the noise level was kept constant. One problem with this method is that of choosing the values of the noise variance σ^2. If a fairly small value of σ is used, so that the noise level is low, then some of the better detectors may yield little or no errors in the duration of a test. Suppose that σ is chosen to be large enough, so that one of the good detection processes gives a reasonable amount of errors, even in conditions of mild signal distortion. Then this value of σ may not be suitable for testing the poorer detection processes, under harsher conditions, i.e. σ may be large enough to give an error rate of $\frac{m-1}{m}$, under these conditions, where m is the number of signal levels.

When the error rate is at this level, the noise is completely swamping the transmitted signal, and the detector is choosing the data element values randomly. (Note that each of the data element values is assumed to occur with probability $\frac{1}{m}$, so a random detection should give an error rate of $\frac{m-1}{m}$). The average error rate should not exceed $\frac{m-1}{m}$ with these detectors, so an increase in σ will not increase the proportion of errors. Hence it can be seen that, if σ is sufficiently large, two detection processes which normally show different performances, will yield the same error rate. Then no useful comparison of the detectors can be made, at this noise level. It is clear from the above discussion, that a suitable value of σ may not be available, for simulation tests involving channels with widely varying degrees of distortion. It was therefore decided to perform the tests at a fixed error rate.

The error rate chosen for the simulation tests was 0.004. In practice, the majority of data transmission systems operate at error rates which are somewhat lower than this. It has however been demonstrated for a particular transmission channel, that the relative performances of the detectors tested, remain constant over a range of error rates from 10^{-1} to 10^{-4}. (See figures 4.02 to 4.05). It is hoped that this is generally the case, so that the best of two detection processes at an error rate of 0.004, is the one which has the best performance over a wide range of error rates.

From Section 4.04, it may be seen that the accuracy of the results obtained in the simulation tests, increases with the number of errors occurring in the test. For a given accuracy, it may be determined that the number n of data elements transmitted during the test, must be large enough to yield a given number q of errors.

Clearly q = n x error rate

so if the chosen error rate is low, n must be large to give results of a required accuracy. Hence, to perform the simulation tests at low error rates, is very demanding on computer time. The error rate of 0.004 was selected as one which was reasonably low, but which would allow a wide range of results to be obtained, with the computing resources available.

4.04 Confidence Limits

Consider a simulation test in which n data elements are transmitted. Let the number of errors in the detected data sequence be q. Then the error rate is defined to be q/n. The expected value e, of the error rate (or the expected error rate), may be defined by

$$e = \lim_{n \to \infty} \frac{q}{n}$$

In a simulation test, n will of course be finite, so the proportion of errors occurring in a test, will only give an estimate of the expected error rate e. Clearly it is necessary to know, at least roughly, how good an estimate of e is being obtained, if one is to have any confidence in the results of the simulation tests. The data necessary to determine the exact accuracy of this estimate of e, has not been obtained for Systems 1-4 or the V.A. This data has however been obtained by J D Harvey, for the decision feedback equalizer described in Section 1.16. As this detection process has a performance which approaches that of a maximum likelihood detector, the distribution of errors it yields should be roughly the same as that for the V.A. detector. Also, where the performances of Systems 1-4 are close to that of the V.A., the distributions of errors should be similar for these systems and the decision feedback equalizer of Section 1.16.

With this decision feedback equalizer, it was found that errors usually occurred in bursts. This implies that, if an error occurs at some point in the detected data sequence, the likelihood of the next few detected elements being in error, is relatively high. Hence the errors in the detected sequence are not statistically independent. It will be assumed, however, that any two error bursts are statistically independent, if g+1 or more data elements are detected correctly between the bursts. (g+1 is the number of components of the sampled impulse response of the channel). Suppose that, at some stage in the detection process, the previous g+1 data elements detected, are all correct. Then the probability of a burst of errors beginning at the detection of the following element, is the probability that this element is detected incorrectly. Let this probability be denoted p. Also, let η be the average number of errors to a burst. Then, during a simulation test in which many data elements are detected, the expected proportion of errors is $p\eta$. (i.e. the proportion of errors is a random variable with mean equal to $p\eta$). If the error rate is low, then at the detection of most of the elements, the previous g+1 data elements will have been correctly detected. Hence, at almost any stage of the detection process, the probability of an error burst beginning is p.

Now consider a simulation test in which n data elements are detected, where n is large. Assume that the expected error rate is low. Then, at the detection of each of approximately n data elements, there is a probability p of an error burst beginning. The process may then be considered as one consisting of approximately n statistically independent experiments (or trials), each of which has the two possible outcomes:

a) an error burst begins

b) an error burst does not begin.

Each of these trials is called a Bernoulli Trial.

Suppose that in a group of n such trials, r successes are recorded, where a success is defined to be the outcome (a).

$$\text{Let } p_1 = r/n \tag{4.01}$$

Then p_1 gives an estimate of the probability p, that the outcome of a particular event is a success. It may be shown (see Appendix 4) that there is a 95% probability, that p is confined to an interval whose lower and upper bounds are approximately

$$p_1 - \frac{2\,p_1}{\sqrt{r}} \quad \text{and} \quad p_1 + \frac{2\,p_1}{\sqrt{r}}$$

i.e.

$$\text{Prob. } \left(p_1 - \frac{2\,p_1}{\sqrt{r}} \le p \le p_1 + \frac{2\,p_1}{\sqrt{r}}\right) = 0.95 \tag{4.02}$$

Then

$$\left(p_1 - \frac{2\,p_1}{\sqrt{r}}\,,\quad p_1 + \frac{2\,p_1}{\sqrt{r}}\right)$$

is called a 95% confidence interval for p. The confidence limits on the estimate of p are $\pm 2\,p_1/\sqrt{r}$.

The average number of errors occurring in a burst has been denoted η, and the number of errors occurring in a simulation test is denoted q. Hence the number of error bursts occurring is given by

$$r = \frac{q}{\eta} \qquad (4.03)$$

Hence equation 4.02 gives

$$\text{Prob. } (p_1 - 2\ p_1 \sqrt{\frac{\eta}{q}} \leq p \leq p_1 + 2\ p_1 \sqrt{\frac{\eta}{q}}) = 0.95$$

or

$$\text{Prob. } (p_1\eta - 2\ p_1\eta \sqrt{\frac{\eta}{q}} \leq p\eta \leq p_1\eta + 2\ p_1\eta \sqrt{\frac{\eta}{q}}) = 0.95 \qquad (4.04)$$

(multiplying throughout by η).

Now let e_1 be the proportion of errors occurring in the simulation test. Then

$$e_1 = q/n$$

where q is the number of errors occurring and n is the number of data elements detected.

Now, from equations 4.01 and 4.03,

$$p_1 = \frac{q}{\eta n} \qquad (4.05)$$

$$\therefore \quad e_1 = p_1\eta.$$

Let e be the expected proportion of errors occurring in the simulation test. p is the expected proportion of error bursts, and the average number of errors per burst is η. Therefore

$$e = p\eta. \qquad (4.06)$$

Applying equations 4.05 and 4.06 to 4.04 gives

$$\text{Prob.}\ (e_1 - 2\ e_1 \sqrt{\frac{\eta}{q}} \leq e \leq\ e_1 + 2\ e_1 \sqrt{\frac{\eta}{q}}) = 0.95$$

Hence the 95% confidence limits for e are

$$\pm 2\ e_1 \sqrt{\frac{\eta}{q}}\ .$$

Clearly the confidence limits are inversely proportional to $\sqrt{q}$.

The noise level chosen for the simulation tests was such that the proportion e_1 of errors was approximately equal to 0.004. At least 60,000 data elements were detected in each test, so the number q of errors per test may be taken as

$$60000 \times 0.004$$

or 240.

Hence the 95% confidence limits for e are given by

$$c = \pm 2 \times 0.004 \sqrt{\frac{\eta}{240}} \qquad \text{or}$$

$$c = \pm 5.164 \times 10^{-4} \sqrt{\eta}\ . \qquad (4.07)$$

The results obtained for the decision feedback equalizer described in Section 1.16, indicate that the average number of errors per burst is about 5, for one of the channels tested. (Channel E in table 4.01). The V.A. detector usually has a slightly better performance than this decision feedback equalizer. It therefore seems reasonable to assume that η will be no greater than 5, for the V.A. detector with channel E. However, some of the channels tested introduce a more severe distortion of the transmitted signal than does channel E. With all of these factors in mind, an average value of 6 for η, seems acceptable for the channels tested, when the V.A. detector is employed. The value $\eta = 6$ also seems to be a good estimate for Systems 1-4, for cases where their performance is close to that of the V.A.

It may be seen from Tables 4.05 and 4.06 that Systems 1-4 have a close to optimum performance, for the channels tested, when the number k of vectors stored at the start of each cycle, is 8 or 16. The performance of these systems is, however, considerably reduced when k is reduced to a value of 4. It has been observed that, with this value of k, the error bursts are generally longer than they are when k takes the values 8 or 16. It was therefore decided to use the value 12, for the average length of an error burst, in cases where k = 4. Hence, with an error rate in the region of 0.004, the confidence limits on the estimate of the error rate are given by

$$c = \left(\begin{array}{ll} \pm 1.789 \times 10^{-3} & \text{for } k = 4 \\ \pm 1.265 \times 10^{-3} & \text{for } k = 8 \text{ or } 16 \end{array} \right. \qquad (4.08)$$

(using equation 4.07).

Now consider a simulation test in which a value σ_1 is chosen, for the noise standard deviation, which gives an error rate of e_1. Let e_1 be close to 0.004 but not actually taking this value. Then, if the gradient of the appropriate curve of error rate against signal to noise ratio is known, the value of σ corresponding to an expected error rate of 0.004, can be calculated as follows:

Assume that the received signal has unit average power, when there is no noise present in the system. Then the signal to noise ratio (S/N ratio), in the received signal may be defined by

$$S/N = 20 \log_{10} (\frac{1}{\sigma}) \text{ db.}$$

Figure 4.01 gives a sketch of a typical plot of expected error rate against S/N ratio.

Let σ_2 be the value of σ (to be determined), which will give an average error rate of 0.004.

Then

$$g_{0.004} \simeq \frac{0.004 - e}{20 \log_{10} (\frac{1}{\sigma_2}) - 20 \log_{10} (\frac{1}{\sigma_1})}$$

where $g_{0.004}$ is the gradient of the curve at an error rate of 0.004, and e is the expected value of the error rate corresponding to a value $\sigma = \sigma_1$. (Note that the error rate corresponding to σ_1, which actually occurred in the simulation test, has been denoted e_1). Then the required value σ_2 is then given by

$$20 \log_{10} (\frac{1}{\sigma_2}) = \frac{0.004 - e}{g_{0.004}} + 20 \log_{10} (\frac{1}{\sigma_1}) \qquad (4.09)$$

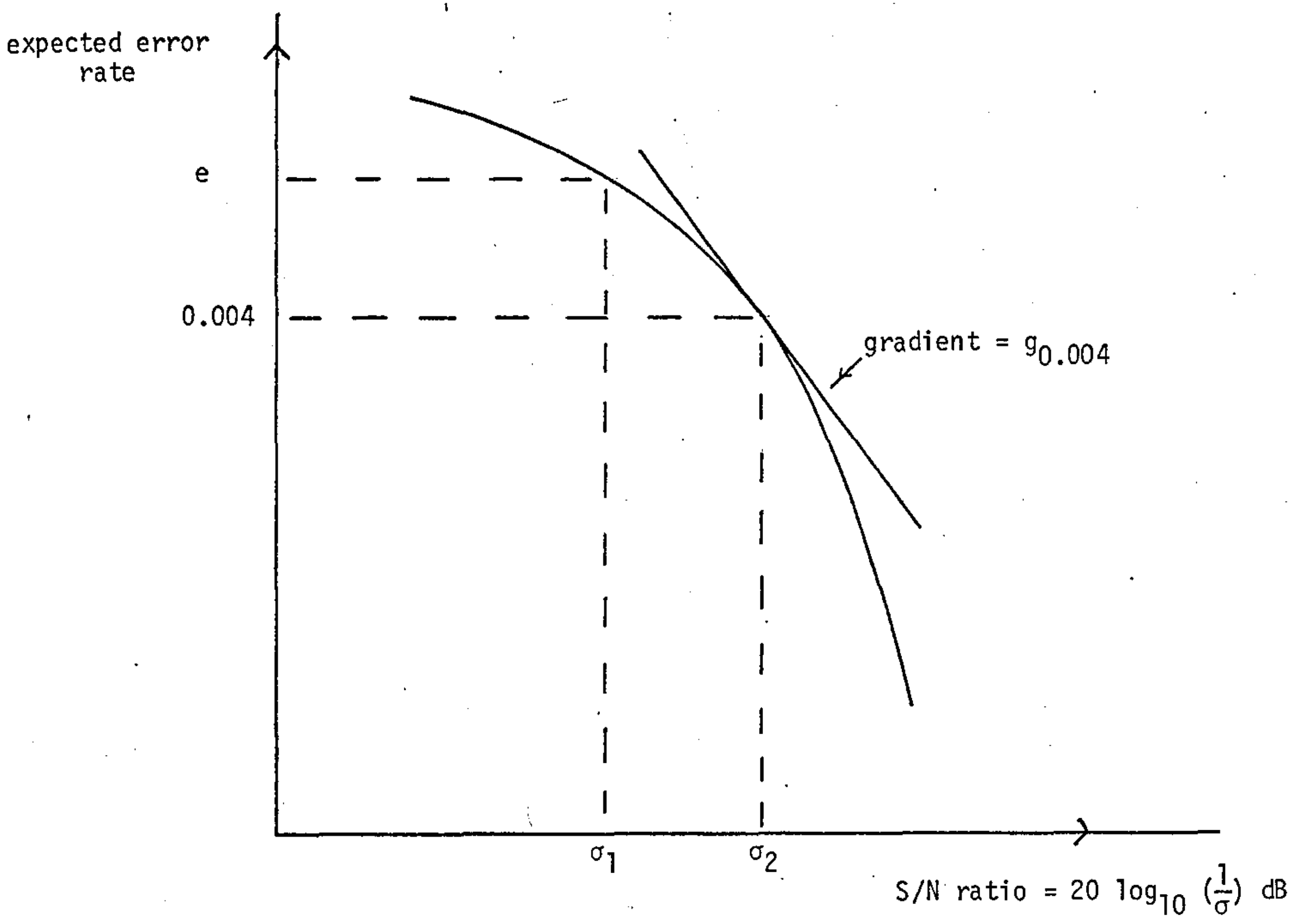
e is not known exactly, but it is known that

$$\text{Prob. } (e_1 - 2 e_1 \sqrt{\frac{n}{q}} \leq e \leq e_1 + 2 e_1 \sqrt{\frac{n}{q}}) = 0.95,$$

as the 95% confidence limits for e_1 are

$$\pm 2 e_1 \sqrt{\frac{n}{q}} .$$

It is therefore possible to find a 95% confidence interval for the value of



Sketch of a curve of expected error rate against signal to noise ratio

FIGURE 4.01

$$20 \log_{10} (\frac{1}{\sigma_2}).$$

The gradient of the curve of error rate against S/N ratio will not normally be known. This curve has, however, been plotted for channel E, with System 1 and k = 4 (see Figure 4.02), and

$$g_{0.004} = -0.003$$

for this case. If e_1 is fairly close to 0.004, a small change in the value of $g_{0.004}$, should not significiantly affect the value of σ_2 given by equation 4.08. Hence it should be possible to obtain a reasonable approximation to

$$20 \log_{10} (\frac{1}{\sigma_2})$$

for most channels, by taking

$$g_{0.004} = -0.003.$$

Hence, from equation 4.09

$$20 \log_{10} (\frac{1}{\sigma_2}) \simeq \frac{e - 0.004}{0.003} + 20 \log_{10} (\frac{1}{\sigma_1}) \qquad (4.10)$$

It has been shown that there is a 95% probability that e lies in the interval

$$(e_1 - c, e_1 + c)$$

where c is given by equation 4.08. Hence, from equation 4.10, there is a 95% probability that

$$20 \log_{10} (\frac{1}{\sigma_2})$$

lies in the interval whose limits are

$$\frac{e_1 \pm c - 0.004}{0.003} + 20 \log_{10} (\frac{1}{\sigma_1})$$

Therefore the 95% confidence limits for

$$20 \log_{10} (\frac{1}{\sigma_2})$$

are $\pm \frac{c}{0.003} \simeq \begin{cases} \pm \text{ 0.6 db for } k = 4 \\ \pm \text{ 0.4 db for } k = 8 \text{ or } 16. \end{cases}$

4.05 Method for Choosing the Noise Level in the Simulation Tests

The purpose of the simulation tests is to find a value σ_1, for the noise standard deviation, such that the resulting error rate e_1 is close to 0.004. Then equation 4.10 may be used to give an estimate of the value σ, which gives an expected error rate of 0.004.

Each simulation test for estimating the tolerance to noise of a system, involved the detection of at least 60,000 data elements. A fairly straightforward way of conducting the tests, is to begin with an initial guess for σ, and run the program for 60,000 data elements, with σ fixed at this value. This method has the disadvantage that, if the guess for σ is a poor one, the resulting error rate will lie a long way from 0.004. Then the estimate of σ_2 given by equation 4.10, will not be very accurate.

To overcome the disadvantage mentioned above, the simulation tests were split into three sections, each covering 20,000 data elements. Then, if the initial guess for σ does not give an error rate which is close to 0.004, for the first section of the test, it may be adjusted for the next section. Then the results of a simulation test consist of three error rates, corresponding to three values for the noise standard deviation σ. A section of a curve of error rate against σ, may then be drawn, and the value of σ corresponding to $e = 0.004$ can be read off.

Strictly speaking, the evaluation of the confidence limits given in section 4.04, is not applicable if the simulation tests are split into three stages, as described above. It is however hoped, that the derived confidence limits will give a reasonable estimate of those appropriate to these tests.

4.06 The Channels Used in the Simulation Tests

The various detection processes were tested over a range of transmission channels with fairly widely varying characteristics, so that the systems which are best for general use could be selected. The sampled impulse responses of the channels are given in Table 4.01.

It was pointed out in section 1.11 that the channels which introduce the greatest degree of amplitude distortion, are usually the ones which give the poorest tolerance to additive white Gaussian noise. The quantity d, defined by equation 1.16, gives a measure of the degree of amplitude distortion caused by the channels. Hence the value of d should give a guide to the tolerance to noise, which results with a channel. Table 4.02 gives the d factor for channels A-L.

TABLE 4.01

Sampled impulse responses of the channels used in the simulation tests

Channel	Sampled impulse responses								
A	0.236	0.943	0.236						
B	0.880	0.471	0.063						
C	0.408	0.816	0.408						
D	0.167	0.500	0.667	0.500	0.167				
E	0.167	0.471	0.707	0.471	0.167				
F	0.319	0.620	0.634	0.323	0.087				
G	0.070	0.478	0.730	0.478	0.070				
H	0.351	0.708	0.591	0.162	0.014				
I	0.085	0.289	0.493	0.577	0.493	0.289	0.085		
J	0.049	0.178	0.338	0.467	0.516	0.467	0.338	0.178	0.049
K	0.548	0.789	0.273	-0.044	0.012	0.017	-0.017	0.007	0.009
L	0.092	0.288	0.507	0.585	0.480	0.266	0.067	-0.034	-0.041	-0.003
	0.023	0.021	0.002	-0.011	-0.011					
M	$\frac{y_0}{\alpha}$	$\frac{0.236}{\alpha}$	$\frac{0.943}{\alpha}$	$\frac{0.236}{\alpha}$						
N	$\frac{y_0}{\alpha}$	$\frac{0.167}{\alpha}$	$\frac{0.500}{\alpha}$	$\frac{0.667}{\alpha}$	$\frac{0.500}{\alpha}$	$\frac{0.167}{\alpha}$				
O	$\frac{y_0}{\alpha}$	$\frac{0.070}{\alpha}$	$\frac{0.478}{\alpha}$	$\frac{0.730}{\alpha}$	$\frac{0.478}{\alpha}$	$\frac{0.070}{\alpha}$				
P	$\frac{y_0}{\alpha}$	$\frac{0.049}{\alpha}$	$\frac{0.178}{\alpha}$	$\frac{0.338}{\alpha}$	$\frac{0.467}{\alpha}$	$\frac{0.516}{\alpha}$	$\frac{0.467}{\alpha}$	$\frac{0.338}{\alpha}$	$\frac{0.178}{\alpha}$	$\frac{0.049}{\alpha}$

$\alpha \triangleq \sqrt{1 + y_0^2}$

Channel	Value of d
A	0.50
B	0.50
C	0.83
D	1.50
E	1.47
F	1.47
G	1.17
H	1.17
I	2.17
J	2.83
K	0.80
L	2.06

TABLE 4.02

The d factor for channels A-L

Channels A, C, D, E, G, I and J are all symmetric in the sense described in section 1.11, so these channels represent pure amplitude distortion in varying degrees. Channels B, F, H, K and L are not symmetric, so they introduce both amplitude, and phase distortion.

If certain assumptions are made about the channels, their frequency characteristics can be determined from their corresponding sampled impulse responses, as shown in Appendix 2. Let the modulus and argument of a channel's frequency response be denoted A(f) and P(f) respectively. Tables 4.03 and 4.04 give a number of equally spaced samples, from the graphs of A(f) and P(f), plotted against f. From the simulation results given later in this chapter, it may be seen that the channels for which the A(f) curves are flattest at the peaks, are the ones which give the best tolerance to additive white Gaussian noise.

The sampled impulse responses of each of the channels A to P, given in Table 4.01, satisfy the condition that the sum of the squares of their components is equal to unity. It will now be shown that, with this condition satisfied and the absence of noise in the system, the average energy of a received signal sample r_k, is the same as that of the data elements s_k.

From equation 1.09, the received signal samples r_k are given by

$$r_k = \sum_{h=0}^{g} s_{k-h} \, y_h + w_k$$

where $\{s_k\}$ and $\{w_k\}$ are the data and noise sequences, and

$$(y_0, y_1, \ldots\ldots, y_g)$$

Channel	$A(-B)$	$A(\frac{-4B}{5})$	$A(\frac{-3B}{5})$	$A(\frac{-2B}{5})$	$A(\frac{-B}{5})$	$A(0)$	$A(\frac{B}{5})$	$A(\frac{2B}{5})$	$A(\frac{3B}{5})$	$A(\frac{4B}{5})$	$A(B)$
A	0.471	0.561	0.797	1.089	1.325	1.415	1.325	1.089	0.797	0.561	0.471
B	0.472	0.562	0.797	1.089	1.324	1.414	1.324	1.089	0.797	0.562	0.472
C	0.000	0.156	0.564	1.068	1.476	1.632	1.476	1.068	0.564	0.156	0.000
D	0.001	0.039	0.088	0.706	1.579	2.001	1.579	0.706	0.088	0.039	0.001
E	0.099	0.048	0.146	0.728	1.572	1.983	1.572	0.728	0.146	0.048	0.099
F	0.097	0.046	0.147	0.729	1.573	1.983	1.573	0.729	0.147	0.046	0.097
G	0.086	0.000	0.321	0.912	1.547	1.826	1.547	0.912	0.321	0.000	0.086
H	0.086	0.000	0.322	0.913	1.547	1.826	1.547	0.913	0.322	0.000	0.086
I	0.001	0.010	0.058	0.277	1.501	2.311	1.501	0.277	0.058	0.010	0.001
J	0.000	0.000	0.001	0.000	1.291	2.580	1.291	0.000	0.001	0.000	0.000
K	0.056	0.147	0.653	1.093	1.444	1.594	1.444	1.093	0.653	0.147	0.056
L	0.007	0.012	0.022	0.211	1.563	2.231	1.563	0.211	0.022	0.012	0.007

$A(f) = |H(f)|$ where $H(f)$ is the Fourier Transform of the channel's impulse response, and f is the frequency in Hz.

B = Bandwidth of channel, assumed the same for channels A-L.

TABLE 4.03

Samples from the amplitude-frequency characteristics for the channels

Channel	P(-B)	$P(\frac{-4B}{5})$	$P(\frac{-3B}{5})$	$P(\frac{-2B}{5})$	$P(\frac{-B}{5})$	P(0)	$P(\frac{B}{5})$	$P(\frac{2B}{5})$	$P(\frac{3B}{5})$	$P(\frac{4B}{5})$	P(B)
A	3.142	2.513	1.885	1.257	0.628	0.000	-0.628	-1.257	-1.885	-2.513	-3.142
B	0.000	0.396	0.541	0.462	0.257	0.000	-0.257	-0.462	-0.541	-0.396	0.000
C	-	2.513	1.885	1.257	0.628	0.000	-0.628	-1.257	-1.885	-2.513	-
D	0.000	1.885	-2.513	2.513	1.257	0.000	-1.257	-2.513	2.513	-1.885	0.000
E	0.000	-1.257	-2.513	2.513	1.257	0.000	-1.257	-2.513	2.513	1.257	0.000
F	0.000	0.389	2.296	1.901	1.003	0.000	-1.003	-1.901	-2.296	-0.389	0.000
G	3.142	-	-2.513	2.513	1.257	0.000	-1.257	-2.513	2.513	-	-3.142
H	0.000	-	2.283	1.609	0.829	0.000	-0.829	-1.609	-2.283	-	0.000
I	0.000	1.257	2.513	-2.513	1.885	0.0	-1.885	2.513	-2.513	-1.257	0.000
J	-	-	-1.885	-	2.513	0.000	-2.513	-	1.885	-	-
K	0.000	1.527	1.529	0.980	0.493	0.000	-0.493	-0.980	-1.529	-1.527	0.000
L	0.000	0.198	-3.068	-2.540	1.803	0.000	-1.803	2.540	3.068	-0.198	0.000

P(f) = Argument of H(f), where H(f) is the Fourier Transform of the channel's impulse response, and f is the frequency is Hz.

B = Bandwidth of channel, assumed the same for channels A-L.

TABLE 4.04

Samples from the phase-frequency characteristics for the channels is the sampled impulse response of the channel under consideration. In the absence of noise in the system,

$$w_k = 0$$

and

$$r_k^2 = (y_0 s_k + y_1 s_{k-1} + \ldots\ldots + y_g s_{k-g})^2$$

$$= \sum_{i=0}^{g} y_i^2 \; s_{k-i}^2 \; + 2 \sum_{i \neq j} y_i \, y_j \, s_{k-i} \, s_{k-j} \qquad (4.11)$$

where the second summation is taken over all values of i and j from 0 to g, such that $i \neq j$.

Now let E (x_k) denote the average value of a sequence of numbers $\{x_k\}$. Then the average energies of the received signal samples r_k, and the data elements s_k, are $E(r_k^2)$ and $E(s_k^2)$ respectively. The possible values of the data elements are

$$\pm 1, \pm 3, \ldots\ldots, \pm(m-1)$$

where m is the number of signal levels (see Section 1.02), and these values occur with equal probability. Hence the average value $E(s_k)$ of the data elements is zero.

From equation 4.11,

$$E(r_k^2) = \sum_{i=0}^{g} y_i^2 E \; (s_{k-i}^2) + 2 \sum_{i \neq j} y_i \, y_j \, E(s_{k-i} \, s_{k-j})$$

But the elements s_k are assumed to be statistically independent for different values of k (see Section 1.02). Hence

$$\begin{aligned} E(s_{k-i} \, s_{k-j}) &= E(s_{k-i}) \, E(s_{k-j}) \\ &= 0 \qquad\qquad \text{for } i \neq j \end{aligned}$$

$$\therefore \quad E(r_k^2) = \sum_{i=0}^{g} y_i^2 \, E(s_{k-i}^2).$$

But $E(s_k^2)$ is the same for all values of k, so

$$E(r_k^2) = E(s_k^2) \sum_{i=0}^{g} y_i^2.$$

Hence the condition:

$$\sum_{i=0}^{g} y_i^2 = 1$$

ensures that the average energy of the received signal samples, is the same as that of the data elements, if there is no noise in the system. When this is the case, the channel is said to have unit gain.

It can be seen from Table 4.01, that channels M, N, O and P have been formed from channels A, D, G and J respectively, by the addition of the component y_0 at the start of the latter sampled impulse responses. For channels M to P, the tolerance to noise of Systems 1-4 was assessed for varying positive values of y_0. The results of the appropriate simulation tests are given in Section 4.11.

Channels B, F and H may be formed from channels A, E and G respectively, by placing a pure phase equalizer in series with the latter three channels. (This is a linear filter which causes only pure phase distortion). The required equalizers are such that the roots of the z transforms of the former three channels,which have modulus greater than unity, are replaced by their reciprocals to form the latter three channels. It may be shown [6,14] that this type of equalizer does not introduce any amplitude distortion in the received signal, and therefore does not have a correlating effect on the noise samples.

The information content of an m level data element is defined to be $\log_2 m$ bits. Now consider a situation where information is required to be transmitted over a channel at some given rate. It can be seen that each data element of a four level signal, has twice the information content of a data element of a binary signal. Hence twice as many elements per second must be transmitted with the binary signal, as with the quaternary one, if the desired information rate is to be achieved. Now consider the model of a data transmission system being used. (See Section 1.02). Clearly, with this model, the impulse response of the channel must be sampled at the same rate, as that at which the data elements are transmitted. Hence the impulse response must be sampled twice as fast with a binary signal, as with a four level signal, and the sampled impulse response will have more components for the binary case (assuming a fixed information rate). For this reason, most of the channels tested with four level signals, were chosen to have fewer components in their sampled impulse responses, than those tested with binary signals.

4.07 Comparison of Detection Processes

For the reasons given in Section 1.09, the various systems under consideration were compared by means of their tolerance to additive white Gaussian noise. The value σ of the noise standard deviation, which gave an average error rate of 0.004, was found for each combination of detection process and transmission channel, by means of simulation tests.

Let σ^* be the value of the noise standard deviation, which gives an error rate of 0.004 with the detection process under consideration, and the ideal channel, (a channel whose sampled impulse response has just one component, equal to unity). Also, let σ be the noise standard deviation which gives an error rate of 0.004, with this detector and some other channel. Then the reduction in tolerance to noise, when this channel replaces the ideal channel, may be defined by

$$R = 10 \log_{10} \left(\frac{(\sigma^*)^2}{\sigma^2} \right) \text{ db} \tag{4.12}$$

From Sections 1.14, 2.10 and 3.12 it can be seen that the non linear equalizer, the V.A. detector and Systems 1-4, are all equivalent when used with the ideal channel. With this channel, the probability of error for the case of a binary signal is given by

$$P_e = \text{Prob. } (w_i > 1)$$

where w_i is a normally distributed random variable with zero mean. (Note that this probability of error is independent of the number k of vectors stored by Systems 1-4, if the ideal channel is used).

$$\text{Prob. } (w_i > 1)$$

implies that the standard deviation for w_i is 0.3774 (from tables of the normal distribution). Hence

$$\sigma^* = 0.3774$$

for the case of a two level signal which can take the values ±1.

From Sections 1.14, 2.10 and 3.12, it can be seen that the probability of error for the case of a quaternary signal, with possible values ±1 and ±3, is given by

$$P_e = 1.5 \text{ Prob. } (w_i > 1)$$

Hence

$$\sigma^* = 0.3597$$

for the case of a four level signal. The reduction in tolerance to noise, when a given channel replaces the ideal channel, is therefore given by

$$R = 10 \log_{10} \left(\frac{(0.3774)^2}{\sigma^2} \right) \text{ for } m = 2$$

and

$$R = 10 \log_{10} \left(\frac{(0.3597)^2}{\sigma^2} \right) \text{ for } m = 4$$

(see equation 4.12), where σ is the noise standard deviation which gives an error rate of 0.004, with the given channel.

The value of R, for the various combinations of transmission channels and detection processes tested, is given in tables 4.05 and 4.06. The first table contains the results for tests with channels C, D, E, F, I, J, K and L and a two level signal. Results for channels A, B, C, E, F, G, H and K with a quaternary signal, are given in table 4.06. The simulation tests on Systems 3 and 4, the V.A. detector and the non linear equalizer were carried out by J D Harvey. (The non linear equalizer tested, is the one of optimum design described in Section 1.13).

Channel	k = 4 stored vectors				k = 8 stored vectors				k = 16 stored vectors				Viterbi-algorithm detector	Nonlinear Equalizer
	System 1	System 2	System 3	System 4	System 1	System 2	System 3	System 4	System 1	System 2	System 3	System 4		
C	3.1	3.7	2.9	2.5	2.9	3.3	2.9	2.5	2.9	2.9	2.9	2.5	2.5	9.0
D	6.9	6.8	7.2	7.6	6.1	6.1	5.9	6.0	6.0	6.2	5.9	5.7	5.6	17.7
E	6.4	6.6	6.9	7.1	5.8	5.8	5.5	5.7	5.7	5.9	5.4	5.3	5.3	11.7
F	5.9	6.2	5.8	5.8	5.6	5.6	5.3	5.3	5.4	5.5	5.3	5.2	5.2	11.7
I	10.8	11.1	11.0	11.1	9.3	9.4	8.9	11.0	9.3	9.3	8.9	9.2	8.5	24.8
J	14.1	14.8	14.9	14.3	12.6	12.2	12.3	13.3	12.3	12.0	12.0	13.1	12.0	30.1
K	2.7	3.0	2.5	2.4	2.6	2.9	2.5	2.4	2.5	2.7	2.5	2.4	2.5	6.2
L	10.4	10.5	11.1	11.1	9.6	9.4	8.9	11.0	9.4	9.0	8.9	9.2	8.7	23.9

FIGURE 4.05

Decibels reduction in tolerance to additive white Gaussian noise, with binary signals, when the given channel replaces one that introduces no distortion or attenuation

Channel	k = 4 stored vectors				k = 8 stored vectors			k = 16 stored vectors				Viterbi-Algorithm detector	Nonlinear Equalizer
	System 1	System 2	System 3	System 4	System 1	System 2	System 3	System 1	System 2	System 3	System 4		
A	0.9	0.9	0.9	0.9	0.9	0.8	0.9	0.8	0.7	0.7	0.6	0.6	1.4
B	1.0	0.6	0.7	0.6	1.0	0.6	0.7	0.8	0.7	0.7	0.6	0.6	1.4
C	4.9	5.5	5.4	5.3	4.7	4.6	4.1	4.7	4.5	4.0	4.0	4.0	9.7
E	11.4	13.9	13.5	13.5	8.9	10.1	11.3	8.7	9.1	8.6	9.2	8.3	12.1
F	8.4	10.3	10.0	10.7	8.5	8.6	8.2	8.4	8.9	8.0	8.1	8.0	12.1
G	10.4	9.9	10.0	10.3	7.7	8.1	8.9	6.8	6.9	6.8	8.4	6.3	11.3
H	6.8	8.8	8.1	8.6	6.7	6.8	6.4	6.6	6.5	6.2	6.2	6.2	11.3
K	4.2	4.4	3.7	4.0	4.0	4.1	3.5	4.0	4.0	3.5	3.7	-	7.0

FIGURE 4.06

Decibels reduction in tolerance to additive white Gaussian noise, with quaternary signals, when the given channel replaces one that introduces no distortion or attenuation.

For all simulation tests described in this section, the number N of components of the stored vectors, was fixed at eleven. With this value of N, the data element s_i is detected upon the arrival of the received signal sample r_{i+11} at the detector, so that the delay in detection is eleven sampling intervals.

From Section 3.09, it can be seen that the amount of computation required per cycle of the V.A., rises rapidly as g increases, (where g+1 is the number of components in the channel's sampled impulse response). For channel K of table 4.01, g=8 and the number of multiplications required by the V.A. for each data element detected is 4^9, if a four level signal is used. Clearly, for this case, a very large amount of computing time would be required to obtain the simulation result. The entry in table 4.06 has therefore been omitted, for the V.A. detected used with channel K. Also, to keep computing time within reasonable bounds, the simulation test for the V.A. with channel L and a binary signal, was carried out with the last six components of the sampled impulse response ignored. These components are fairly small, so this omission should not greatly affect the tolerance of the system, to additive white Gaussian noise.

It is clear from tables 4.02, 4.05 and 4.06, that the V.A. detectcr offers a considerable improvement in performance over the non linear equalizer, for the channels which introduce severe amplitude distortion. These are the channels which have the highest d rating in table 4.02, and which give the poorest tolerance to additive white Gaussian noise.

With k = 16, (i.e. 16 vectors stored at the start of each cycle), Systems 1-4 offer a performance which is quite close to that of the V.A. detector. The greatest discrepancy then occurring, is between the V.A. and System 4 when used with channel G, and is about 2 db. The loss in performance, when Systems 1-4 are used with k reduced to 4, is quite noticeable for the channels which introduce severe amplitude distortion. System 4 can be seen to have a poorer performance than Systems 1-3, for some of the channels tested, whether k takes the value 4, 8 or 16. This appears to be a penalty that must be paid, for the fact that it requires fewer operations per detected element than Systems 1-3, (see Section 3.09). For k = 4, the channels which introduce severe amplitude distortion, and a four level signal, System 1 seems to offer a slightly better performance than the other three systems.

From the descriptions of Systems 2, 3 and 4, it can be seen that each of them will contain the same stored vectors, for a case where a four level signal is used, and k = 4. (k is the number of vectors stored at the start of each cycle). For this case, the decision rules for each system will ensure that the vector with lowest cost is selected, for each of the four possible values of the latest component of the vectors. Systems 2, 3 and 4 will then produce the same detected data sequences, and the tolerance to noise will be the same for the three systems. It can however be seen from table 4.06, that the tolerance to noise figures do not agree, for Systems 2, 3 and 4 with k = 4. This is because the systems have been tested with separate simulation trials, and the outcome of each trial is subject to statistical fluctuation. For Systems 2, 3 and 4 with k = 4 and a quaternary signal, a more accurate tolerance to noise figure may be obtained, by taking the average of the figures given for the three systems.

It is evident from the tables that when the V.A. detector is used, channels A, E and G yield the same tolerances to noise as channels B, F and H respectively, (within a reasonable tolerance which should be allowed, for statistical fluctuation). But channels B, F and H are obtained from channels A, E and G respectively, by the use of a pure phase equalizer (see Section 4.06). Hence it would appear that there is no advantage to be gained by using this pure phase equalizer, in conjunction with the V.A. detector. This equalizer does, however, offer an improvement in tolerance to noise with Systems 1-4, when used with k = 4 or k = 8, hence supporting the conclusions of Section 3.11.

Clearly with k = 16, Systems 1-4 offer a performance which is quite close to that of the V.A. detector, for the channels tested. (See tables 4.05 and 4.06). The advantage of these systems over the V.A. is, of course, the fact that the number of basic operations required by them per detected data element, is sometimes much less than the number required by the V.A. The difference in the amount of computation required by the V.A. and Systems 1-4, is shown in tables 4.07 and 4.08, which are for two and four level signals, respectively. The number of basic operations (i.e. multiplications, and comparisons between two numbers), may be calculated from the expressions given in Sections 2.09 and 3.09. It can be seen from the tables, that Systems 1-4 do not offer a significant reduction in computation, over the V.A. detector, unless a sampled impulse response with a large number of components is being used. (i.e. unless

Channel	g	V.A.	System 1			System 2			System 3			System 4		
			k=4	k=8	k=16	k=4	k=8	k=16	k=4	k=8	k=16	k=4	k=8	k=16
C	2	12	30	108	408	18	60	216	18	60	216	12	24	48
D	4	48	30	108	408	18	60	216	18	60	216	12	24	48
E	4	48	30	108	408	18	60	216	18	60	216	12	24	48
F	4	48	30	108	408	18	60	216	18	60	216	12	24	48
I	6	192	30	108	408	18	60	216	18	60	216	12	24	48
J	8	768	30	108	408	18	60	216	18	60	216	12	24	48
K	8	768	30	108	408	18	60	216	18	60	216	12	24	48
L	14	49152	30	108	408	18	60	216	18	60	216	12	24	48

TABLE 4.07

Number of multiplications + number of comparisons required for the detection of each data element, with binary signals.

Channel	g	V.A.	System 1			System 2			System 3			System 4	
			k=4	k=8	k=16	k=4	k=8	k=16	k=4	k=8	k=16	k=4	k=16
A	2	112	70	252	952	28	84	280	28	84	280	28	112
B	2	112	70	252	952	28	84	280	28	84	280	28	112
C	2.	112	70	252	952	28	84	280	28	84	280	28	112
E	4	1792	70	252	952	28	84	280	28	84	280	28	112
F	4	1792	70	252	952	28	84	280	28	84	280	28	112
G	4	1792	70	252	952	28	84	280	28	84	280	28	112
H	4	1792	70	252	952	28	84	280	28	84	280	28	112
K	8	458752	70	252	952	28	84	280	28	84	280	28	112

TABLE 4.08

Number of multiplications + number of comparisons required for the detection of each data element, with quaternary signals.

g is greater than about four). It should be noted that the number of operations required by Systems 1-4, is governed by the number of stored vectors employed, and not by the value of g. (i.e. the number of operations is independent of the transmission channel being used).

4.08 Variation of Error Rate with Signal to Noise Ratio

The simulation tests described in Section 4.07, compare the various detection processes, when they are operating at an error rate of 0.004. It is not, however, safe to conclude from these tests alone, that the relative performances of the systems, will be the same at other error rates. Hence, for channel E and a binary data signal, the performances of the various detection processes were examined over a range of error rates. Graphs of error rate against signal to noise ratio, were then produced, for error rates from 10^{-1} to 10^{-4}.

Note that the sampled impulse response of channel E, given in table 4.01, is such that the sum of the squares of its components is unity. (This is true for all of the channels given in table 4.01). This ensures that the average power $E(r_k^2)$ of the received signal, is the same as the average transmitted signal power $E(s_k^2)$. (See Section 4.06). For binary signals, the data elements may take the values ±1, so the average power of the transmitted signal is unity.

The noise samples w_i, at the output of the transmission channel, are assumed to be normally distributed random variables with zero mean and some variance σ^2. Hence

$$E\,[w_i^2] = E\,[(w_i - \mu)^2]$$

$$= \text{var}\,(w_i)$$

$$= \sigma^2$$

where E denotes the expected value, and μ is the expected value of w_i and is equal to zero. The signal to noise ratio is defined by

$$R^* = 10\,\log_{10}\ (\text{signal power/noise power})$$

Hence, in this case where the signal power is unity, the S/N ratio for the received signal, is given by

$$R^* = 10\,\log_{10}\,(\frac{1}{\sigma^2})$$

Note that the quantity R, given in table 4.05, is defined by

$$R = 10\,\log_{10}\,(\,\frac{(0.3774)^2}{\sigma^2}\,)$$

Hence R and R* differ only by the constant additive factor

$$10\,\log_{10}\,(0.3774)^2 = -8.464$$

Figures 4.02 - 4.05 show graphs of error rate against S/N ratio, for the various detection processes, with a binary data signal and channel E. The graphs were obtained by carrying out simulation tests at various S/N ratios (various values of σ), and noting the resulting error rates e. Hence each simulation test

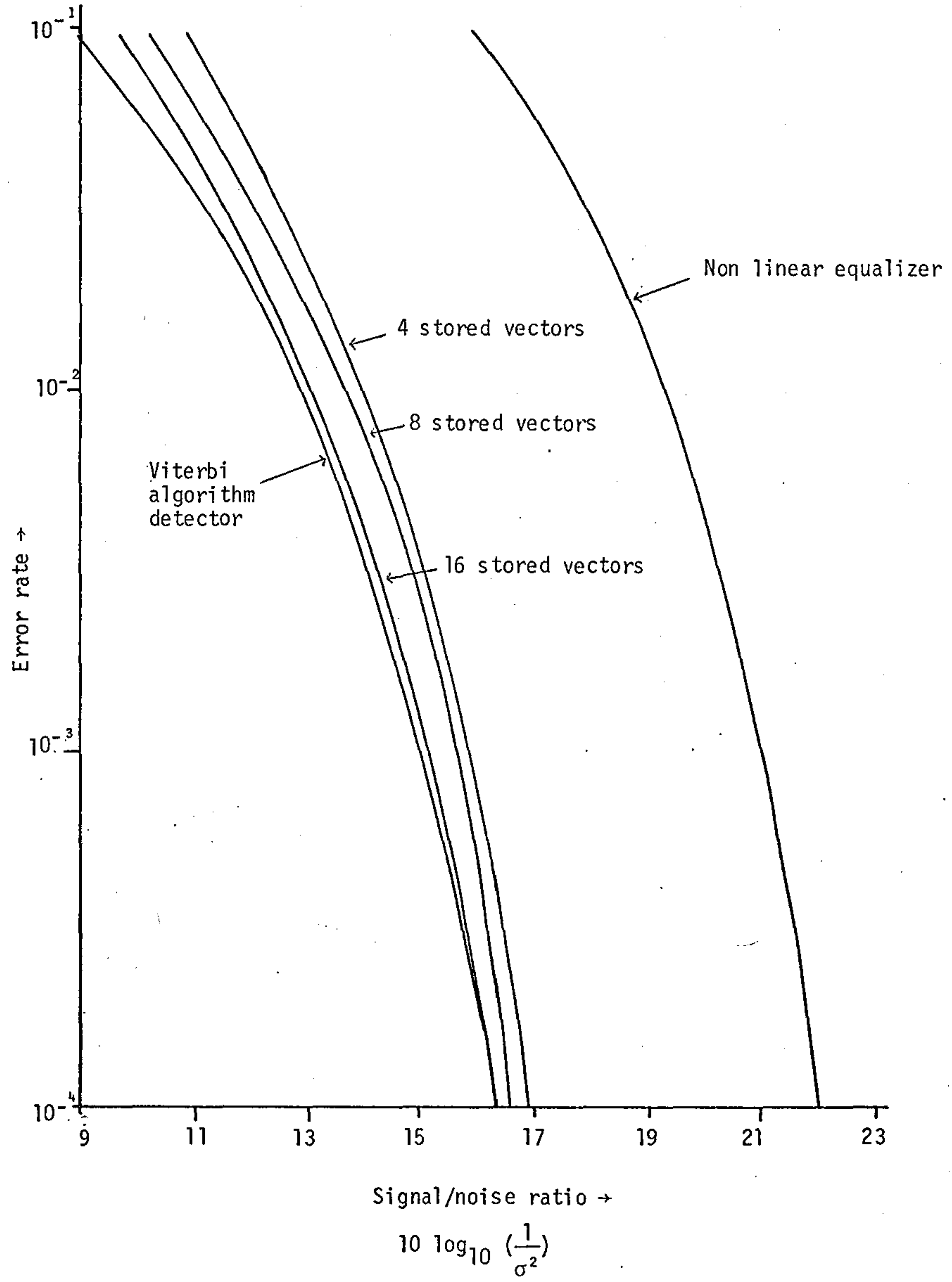


FIGURE 4.02

Variation of error rate with signal to noise ratio for System 1 operating with binary signals over channel E.

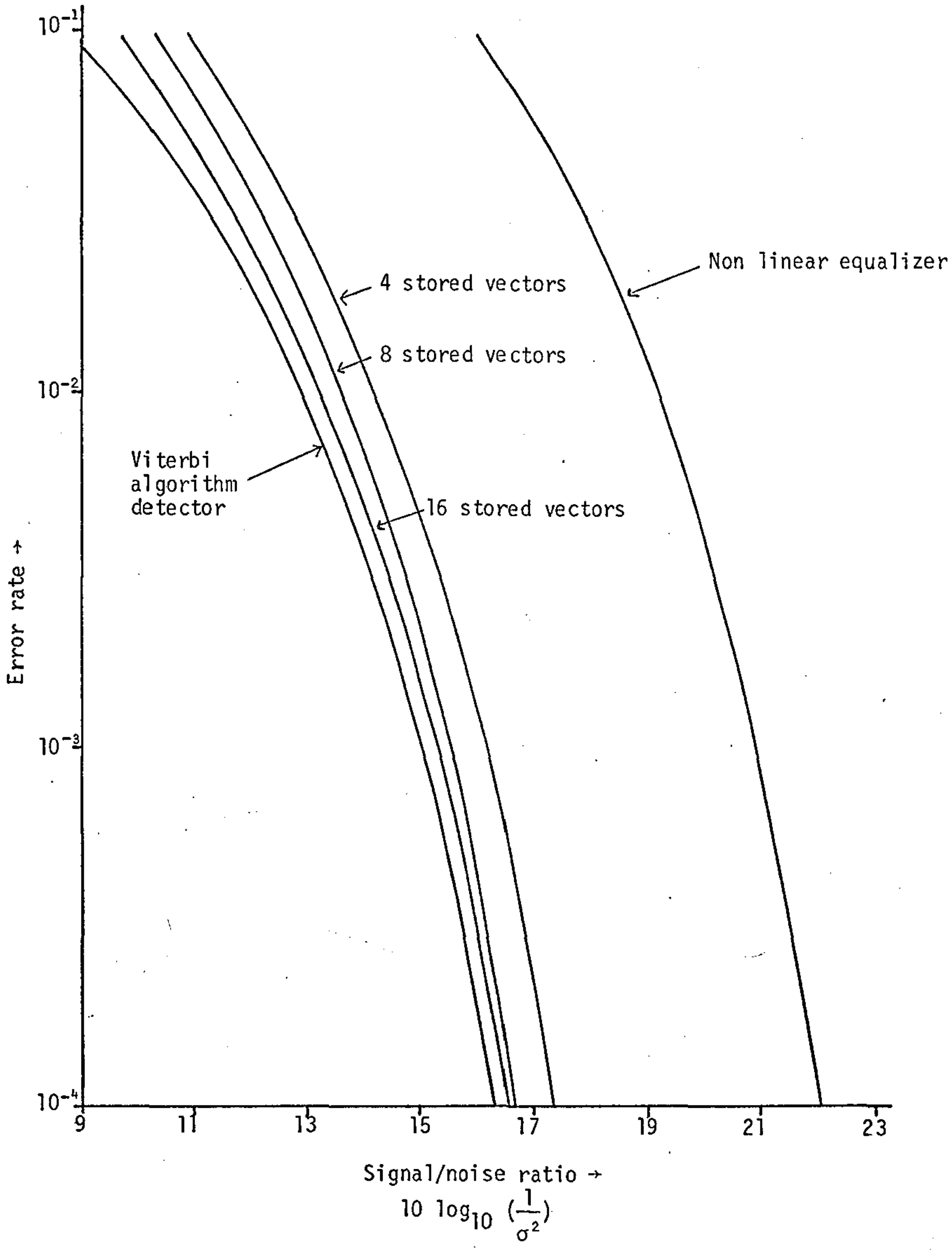
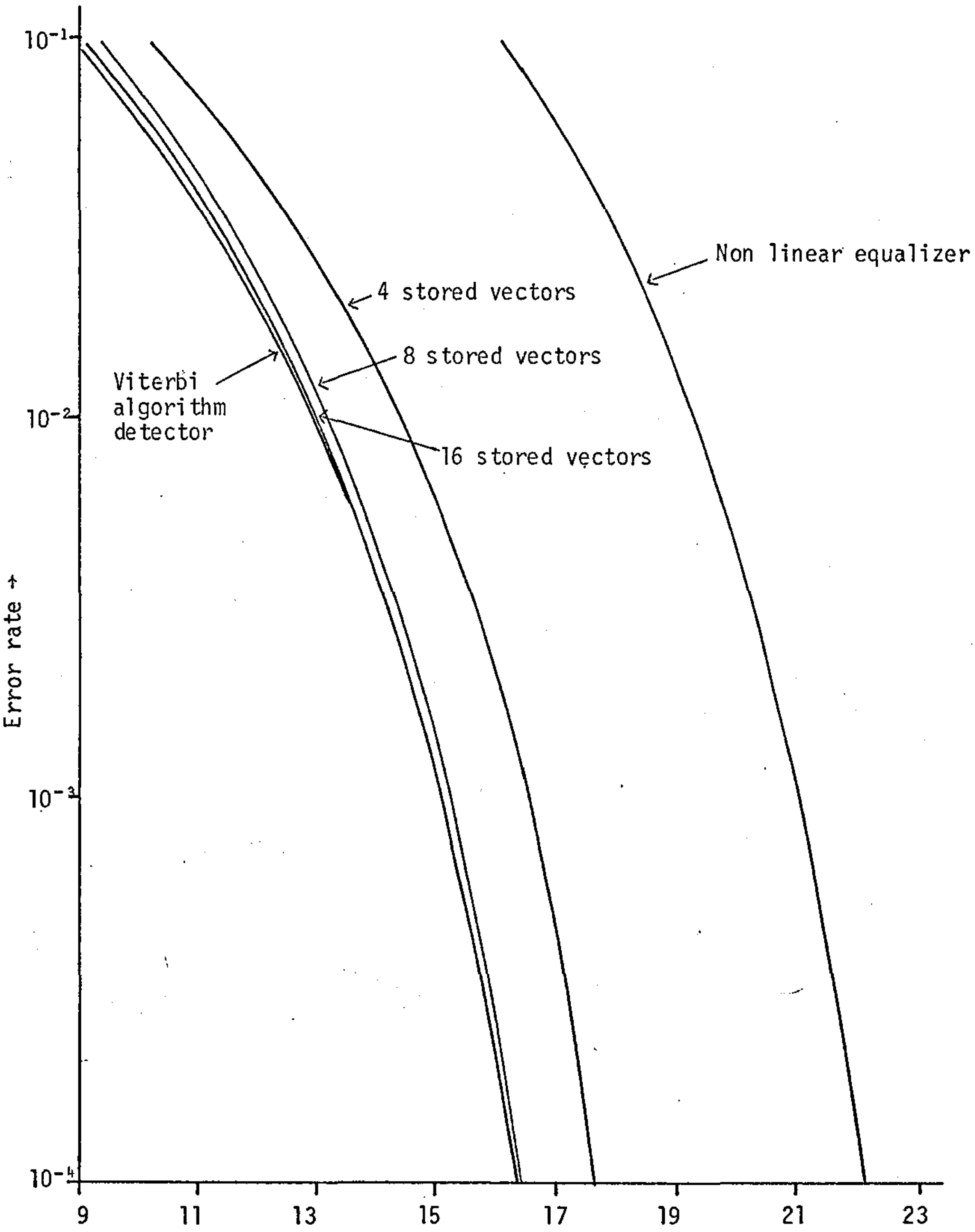


FIGURE 4.03

Variation of error rate with signal to noise ratio for System 2 operating with binary signals over channel E.



Signal/noise ratio →

$10 \log_{10} (\frac{1}{\sigma^2})$

FIGURE 4.04

Variation of error rate with signal to noise ratio for System 3 operating with binary signals over channel E.

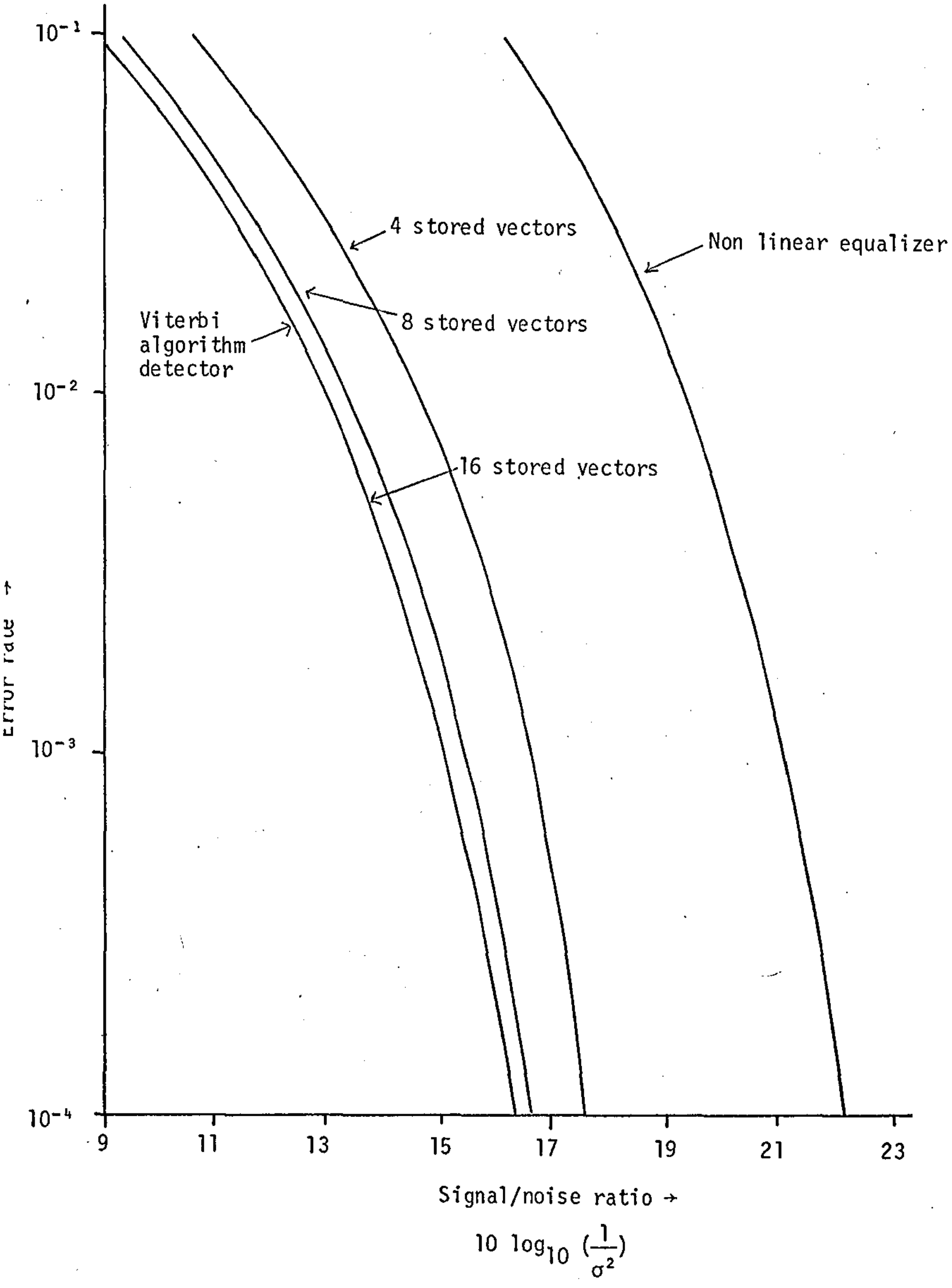


FIGURE 4.05

Variation of error rate with signal to noise ratio for System 4 operating with binary signals over channel E.

provided a pair of coordinates (σ,e). A graph was then drawn through the resulting pairs of coordinates, for each of the systems.

The coordinates (σ,e), obtained in each simulation test, will be subject to statistical fluctuation, as were the results of the tests described in Section 4.07. i.e., if a simulation test was repeated many times with the same value of σ, different values for e would probably occur on each occasion. Each of these values would give an estimate of the expected error rate, corresponding to the given value of σ. The accuracy of these estimates is dependent on the number of errors occcurring in each test. (See Section 4.04). Clearly, at low error rates, a large number of data elements must be detected in each simulation test, if a reasonable number of errors are to occur. In practice, data transmission systems commonly work at error rates as low as 10^{-6}. It would however have required a very large amount of computer time, to obtain coordinates for the graphs, at such error rates. Hence the lowest error rate considered was 10^{-4}.

Figures 4.02 - 4.05 cover Systems 1-4 respectively, and show the performances of the systems for 4, 8 or 16 stored vectors, (i.e. for k = 4, 8 or 16). The number N, of components of the vectors, was fixed at eleven, as for the tests described in Section 4.07. Curves of error rate against S/N ratio, for the V.A. detector and the optimum non linear equalizer described in Section 1.13, are also shown on figures 4.02 - 4.05. Hence the performances of these two detectors can readily be compared with the performances of Systems 1-4.

It can be seen that the V.A. detector maintains a significant advantage over the non linear equalizer, for all error rates considered, with channel E and a binary signal. A comparison of figures 4.02 - 4.05, reveals the relative performances of the various detectors, for error rates from 10^{-1} to 10^{-4}. It can be seen from these figures that the relative performances of the systems, at an error rate of 0.004, is representative of their performances over the full range of error rates.

Figure 4.05 shows that, when System 4 is used with 16 stored vectors (i.e. $k = 16$), it has the same performance as the V.A. detector, in the given situation. This is because the V.A. detector requires 16 stored vectors, for channel E with a binary signal, and is equivalent to System 4 for this case. (Compare the descriptions of the two algorithms, given in Chapters 2 and 3).

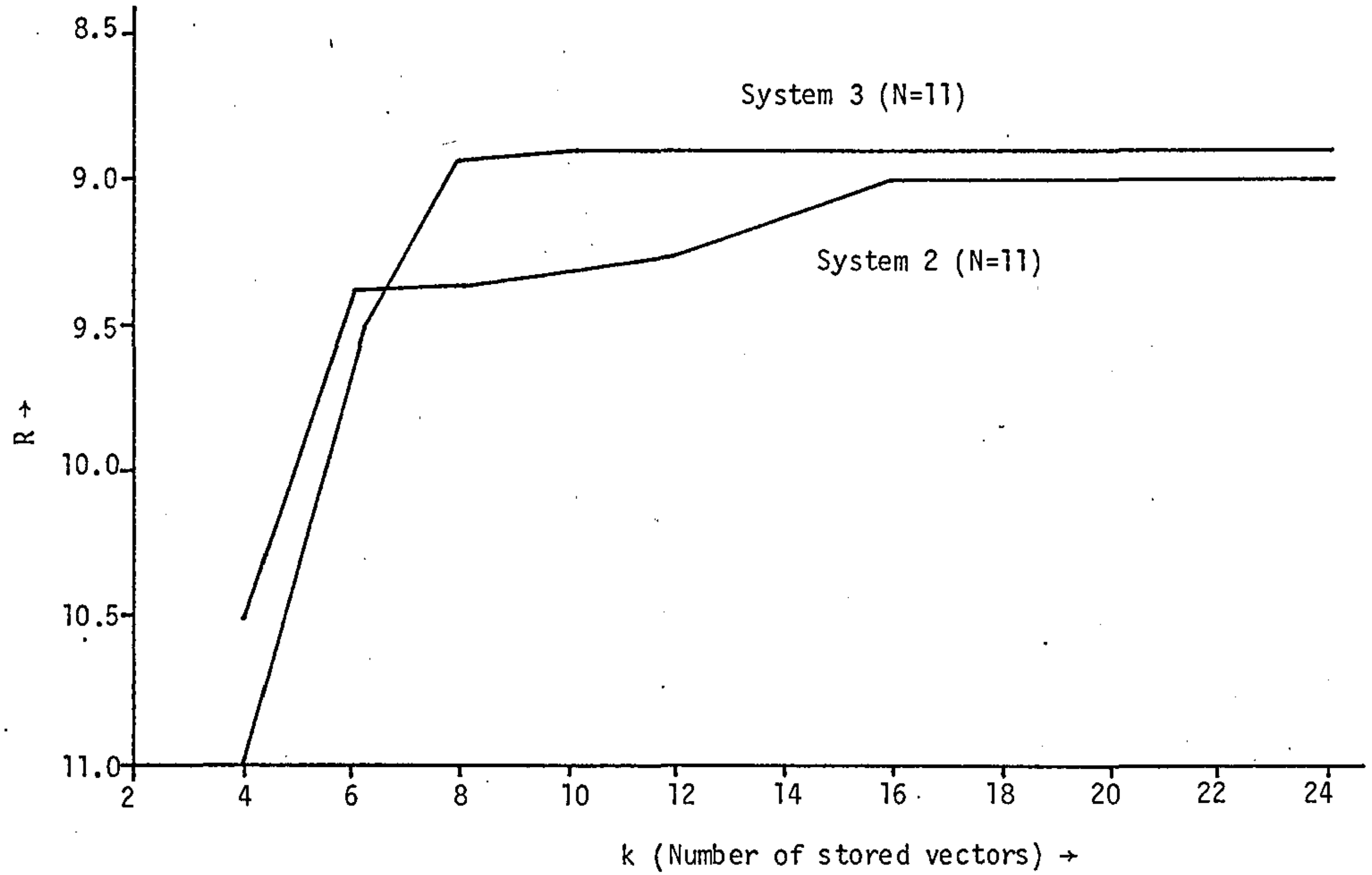
When plotting the graphs for figures 4.02 and 4.03, it was found that a smooth curve could be fitted quite closely to almost all of the points. There were however a few points which were located at a considerable distance from this curve. The error rates in these cases were much higher than expected, suggesting that the detection process had begun to break down in some way. This drop in the performances of Systems 1 and 2, was noticed only in a few of the long simulation tests, which were required for the results at low error rates. The points on the graphs which were situated a long way, from the curve indicated by the vast majority of points, were ignored so that a smooth curve could be plotted. Hence the given curves for Systems 1 and 2, represent their performances when the cases of unusual behaviour have been excluded. This phenomenon in which Systems 1 and 2 can lose performance, during the detection of long data sequences, is discussed at length in Chapter 5.

4.09 Variation in Performance with the Number of Stored Vectors used, for Systems 1-4

Unlike the V.A. detector, Systems 1-4 allow a choice of the number k, of vectors stored at the start of each cycle of the detection process. It is therefore useful to know how the number of vectors used, affects the performances of the systems, and how many vectors are required to ensure a close to optimum tolerance to noise. The variation in performance of the systems, for k = 4, 8 and 16, may be assessed from tables 4.05 and 4.06. In addition, the performances of Systems 2 and 3, with two of the channels from table 4.01, were tested for a wider range of values of k. Figures 4.06 and 4.07 show graphs of reduction in tolerance to noise against k, with these two systems, for channel L with a binary signal and channel K with a quaternary signal. (The reduction in tolerance to noise is expressed by R, given in equation 4.12, as was the case for tables 4.05 and 4.06).

For the simulation tests described in this section, the number N of components of the vectors stored at the start of each cycle, was fixed at eleven. The tests were carried out at an error rate of 0.004, as for those described in Section 4.07.

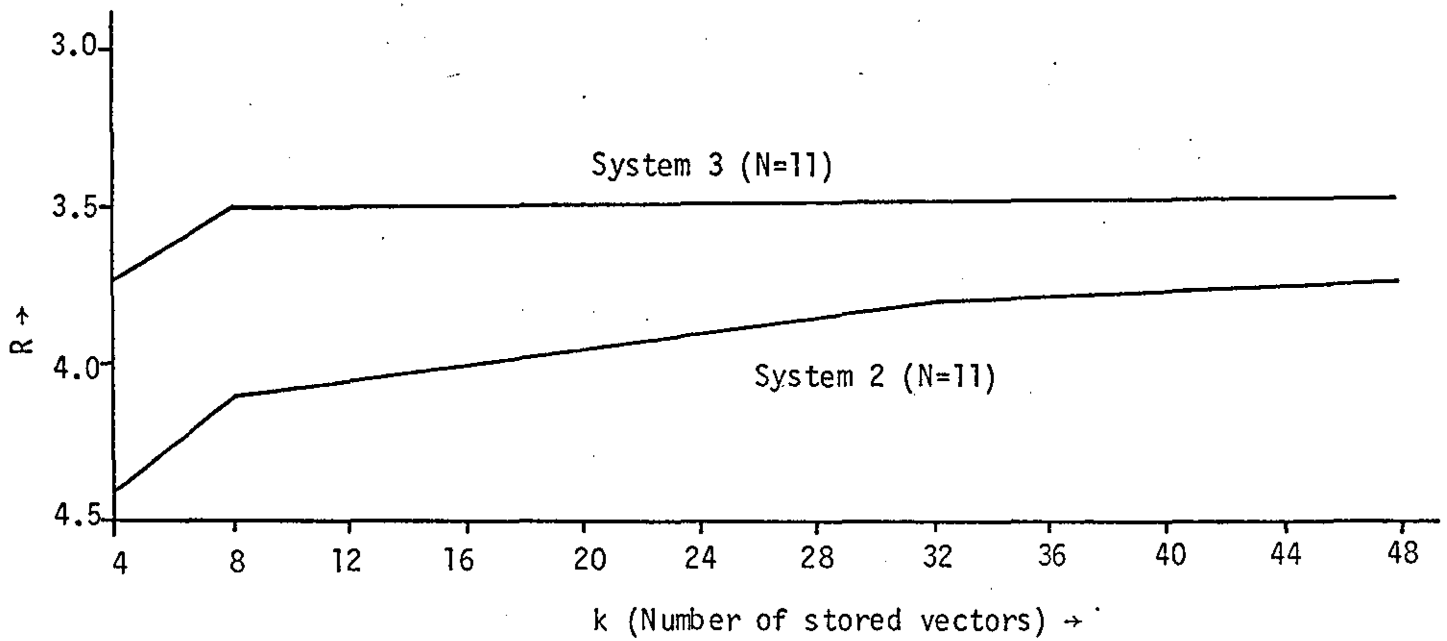
It can be seen from figures 4.06 and 4.07, that the tolerance to noise of Systems 2 and 3, increases rapidly as k increases from four to eight. For System 3 and the cases tested, there is no significant improvement in performance to be had, by increasing k beyond eight. System 2, however, requires a greater number of stored vectors to achieve its best performance, for the given situation.



R is the reduction in tolerance to noise when channel L replaces one that introduces no distortion or attenuation.

FIGURE 4.06

Variation of tolerance to noise with k. Binary signals.



R is the reduction in tolerance to noise when channel K replaces one that introduces no distortion or attenuation.

FIGURE 4.07

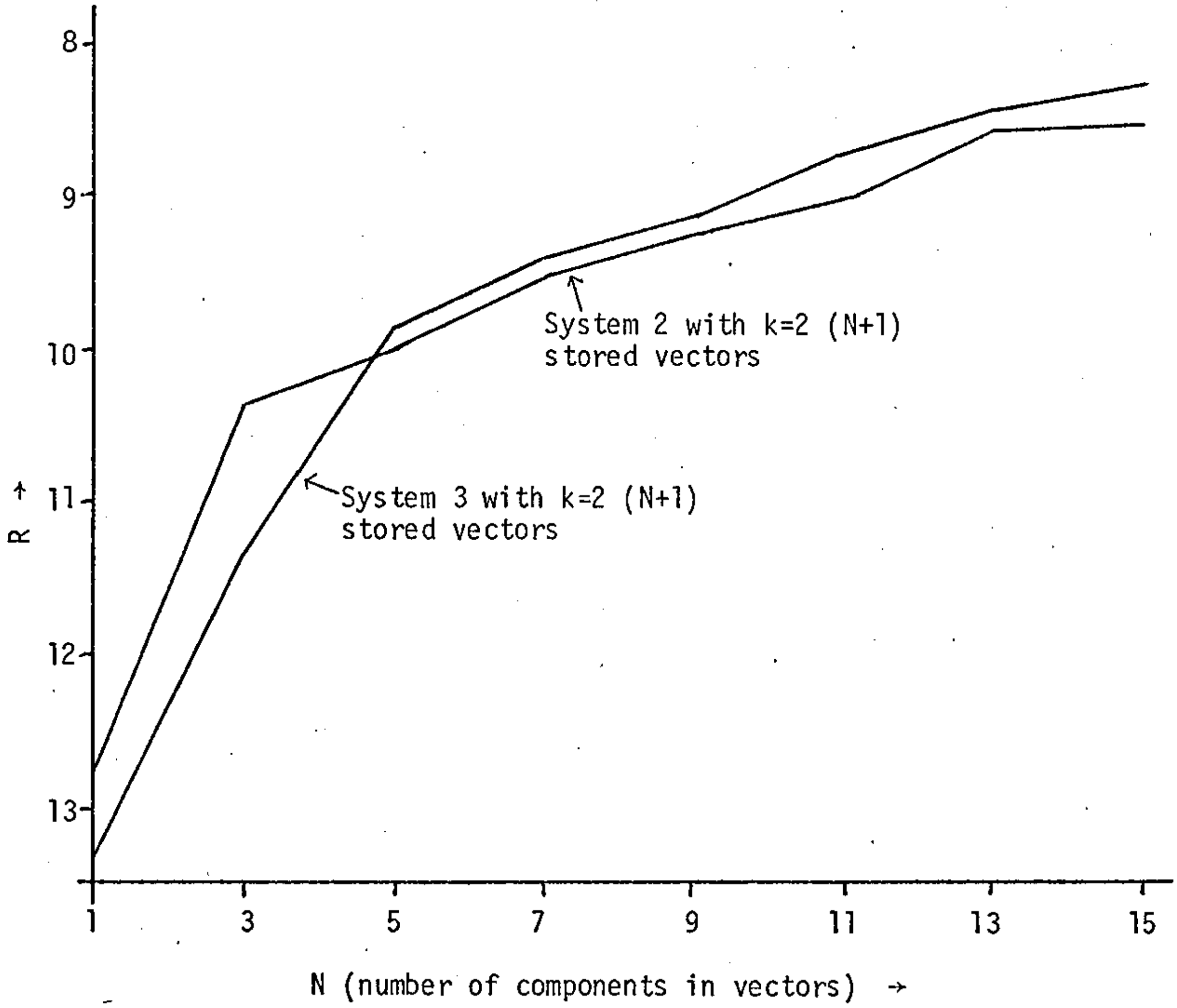
Variation of tolerance to noise with k. Quaternary signals.

4.10 Variation in Performance with Both the Number of Stored Vectors, and the Number of Components of the Vectors

For the simulation tests described in Section 4.09, the number N, of components of the vectors stored at the start of each cycle, was fixed at eleven. It was seen that, with System 3 and the channels tested, there was little improvement to be obtained by increasing the value of k past eight. It may be found, however, that greater values of k may offer an increased tolerance to noise, if a different value of N is used.

For a given value of N, the maximum number k of vectors stored at the start of each cycle of System 3, is m(N+1), where m is the number of signal levels. (See Section 3.06). It was desired to plot graphs of performance against N, for values of N from one upward. Hence, if all of the tests were conducted with a fixed value of k, this value could be at most four for a binary signal, and eight for a four level signal. However, with these values of k, Systems 2 and 3 may not reach their best possible performance, no matter how large the value of N used. It was therefore decided that, for any given value of N, the simulation tests would be conducted with k = m(N+1), thus using the maximum number of stored vectors that is possible with System 3.

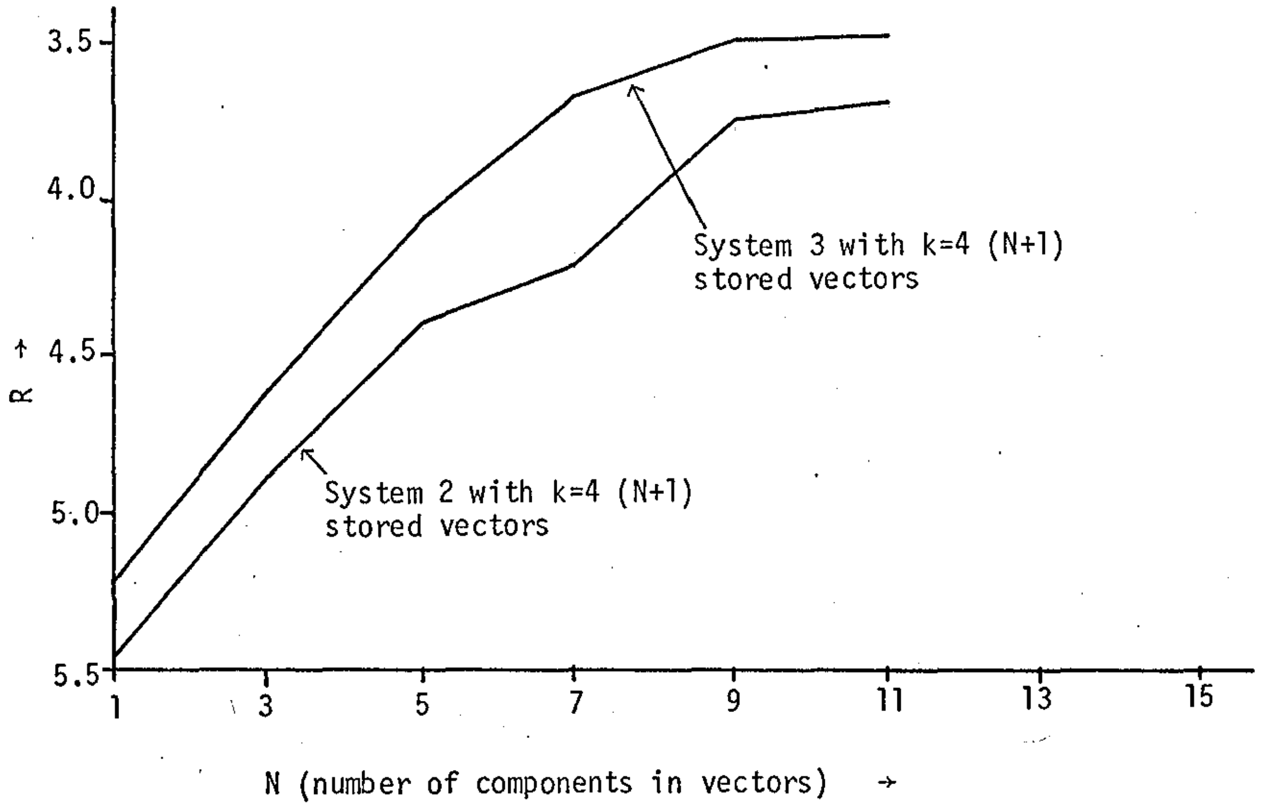
In figures 4.08 and 4.09, graphs are given of performance against N, for Systems 2 and 3. The performances of the systems are specified in terms of R, defined by equation 4.12, as in Sections 4.07 and 4.09. (The value of σ in the expression for R, is of course the value which gives an expected error rate of 0.004). Figure 4.08 covers the case of a binary signal used with channel L, and figure 4.09 is for a quaternary signal used with channel K.



R is the reduction in tolerance to noise, when channel L replaces one that introduces no distortion or attenuation.

FIGURE 4.08

Variation in performance with both N and k. Binary signals.



R is the reduction in tolerance to noise, when channel K replaces one that introduces no distortion or attenuation.

FIGURE 4.09

Variation in performance with both N and k. Quaternary signals.

(Note that channels L and K were the ones used for the tests described in the previous section).

A comparison of figures 4.06 and 4.08 shows that, for a two level signal and channel L, Systems 2 and 3 do not attain their best possible performances with N = 11. For example, with N = 11, the System 2 tolerance to noise figure (i.e. the value of R), does not rise above 9.0 db, no matter how far k is increased. However, with N = 15 and k = 32, a lower value for R is obtained. With channel K and a quaternary signal, it can be seen that a value of eleven for N, is large enough to obtain the best performances of Systems 2 and 3, for the given situation. Hence it may be concluded that there are no fixed minimum values for k and N, which will ensure that the best tolerances to noise are offered by Systems 2 and 3, for all situations.

4.11 The Effect of Ignoring the First Component of the Channel's Sampled Impulse Response

It can be seen from Section 3.10 that, if a zero component at the start of the channel vector is removed, the effect on the performances of Systems 1, 2 and 4 is equivalent to that of increasing k by a factor of m. (m is, of course, the number of signal levels, and k is the number of vectors stored at the start of each cycle of the process). Simulation tests show that this result also holds fairly well, if a small non zero component is removed from the start of the channel's sampled impulse response. If this first component of the channel vector is sufficiently small, it will not make a significant contribution to the received signal. In this case, it may be advantageous if this component is ignored by the detector, even though the actual channel vector must remain unaltered.

Now consider Systems 1-4, when used with a transmission channel whose sampled impulse response is

$$(y_0, y_1, \ldots\ldots, y_g).$$

Define Systems 1A-4A to be the same as Systems 1-4, apart from the fact that the former four detection processes take the channel vector to be

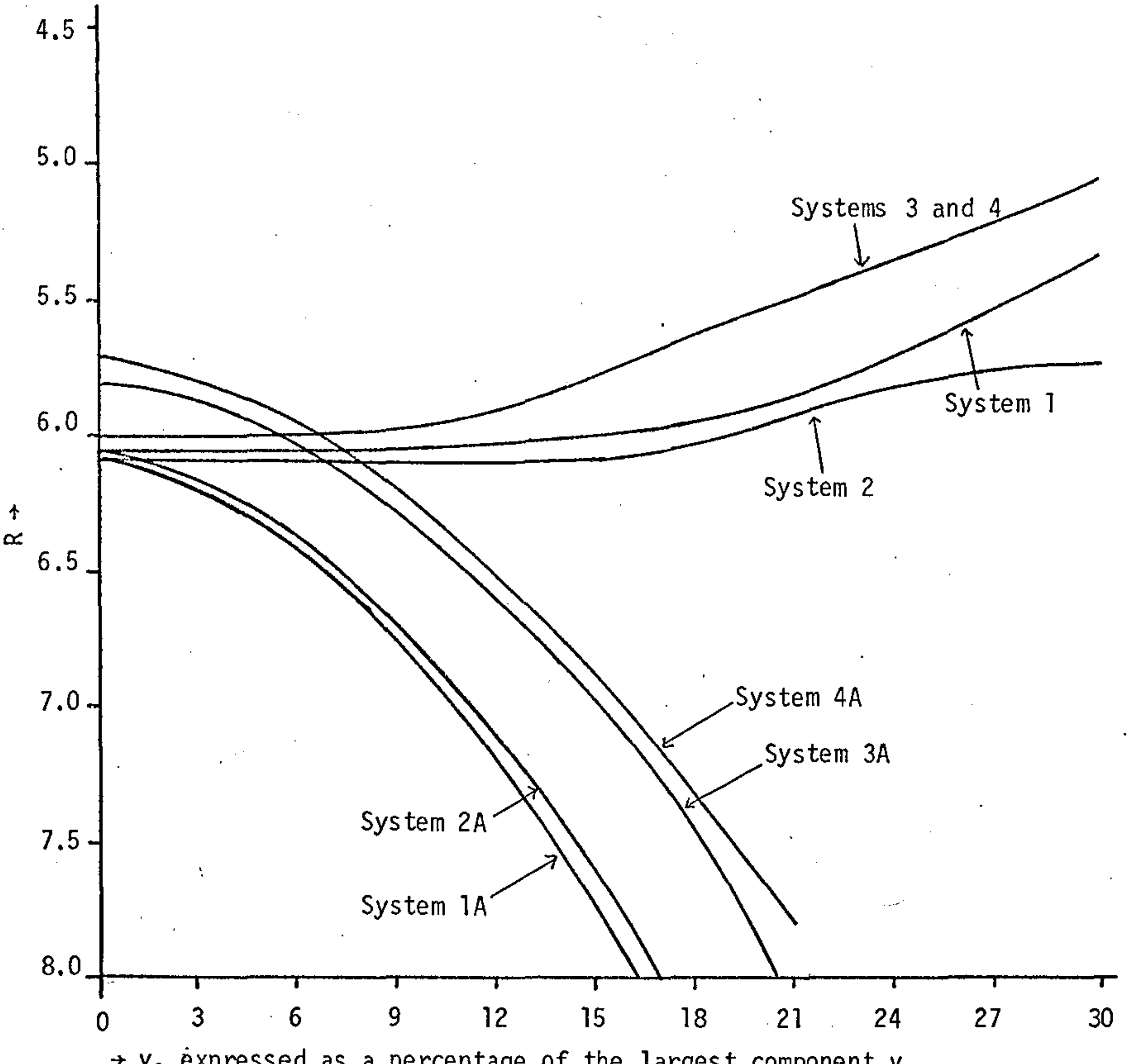
$$(y_1, y_2, \ldots\ldots, y_g)$$

i.e. with Systems 1A-4A, the first channel component is ignored in the detection process.

Simulation tests were carried out on the eight Systems 1A-4A and 1-4, with channels M, N, O and P from table 4.01. The first component y_0, of these channel vectors, was given various non negative values, for these simulation tests. The tests were carried out at an error rate of 0.004 and, as before, the performances of the systems were specified in terms of the quantity R, defined by equation 4.12. Throughout these simulation tests, the number k of stored vectors was fixed at sixteen. The number N, of components of the vectors, was eleven.

Figures 4.10 - 4.13 show graphs of R plotted against y_0, for Systems 1-4 and 1A-4A. The first two figures cover the case of a two level signal used with channels N and P. Figures 4.12 and 4.13 are for a quaternary signal, used with channels M and O respectively.

Now consider a case where the first component y_0, of the channel vector, is equal to zero. Note that channel M is formed

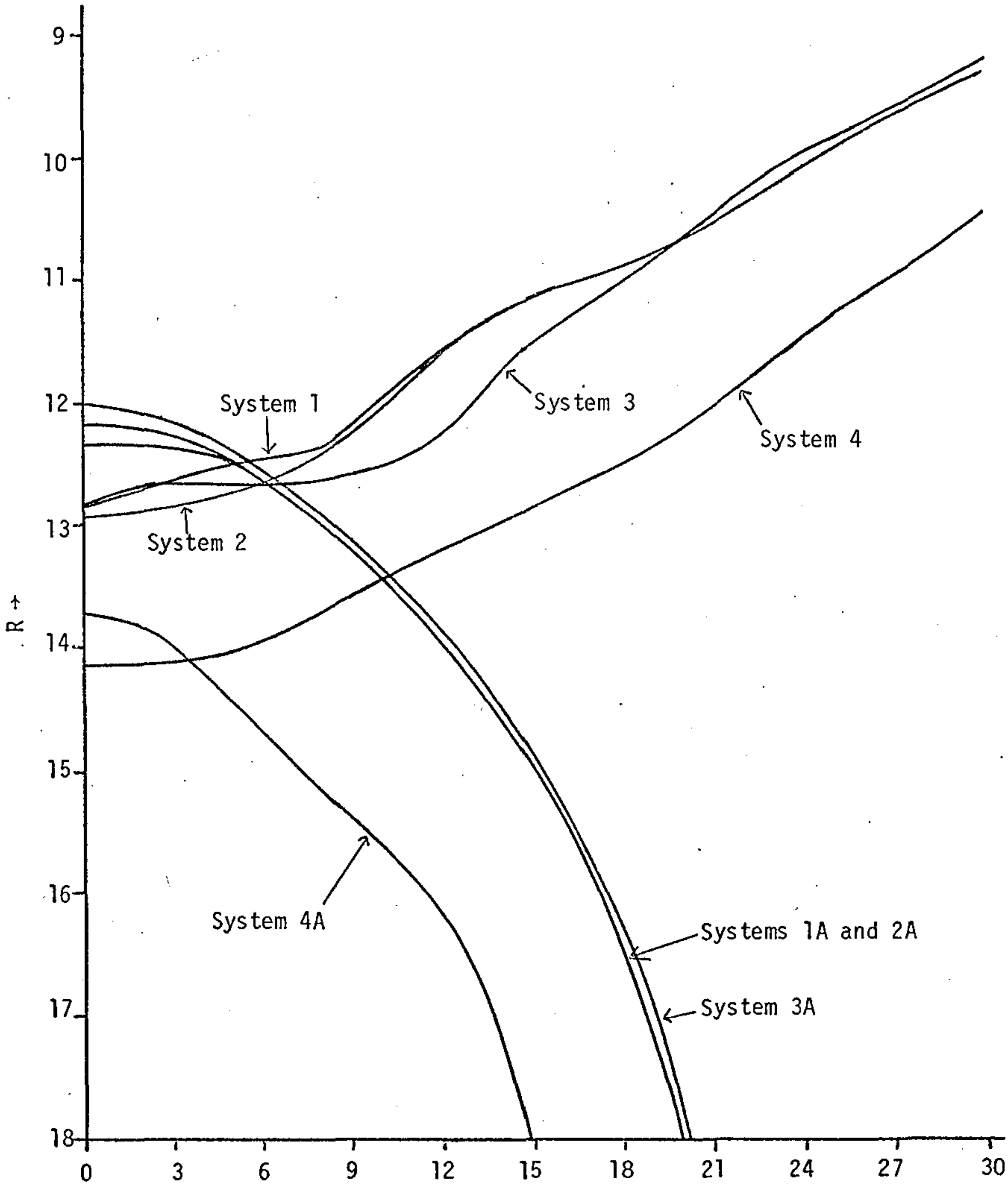


→ y_0 expressed as a percentage of the largest component y_i.

R is the reduction in tolerance to noise, when channel N replaces one which introduces no distortion or attenuation.

FIGURE 4.10

Variation in performance with y_0. Binary signals with channel N.

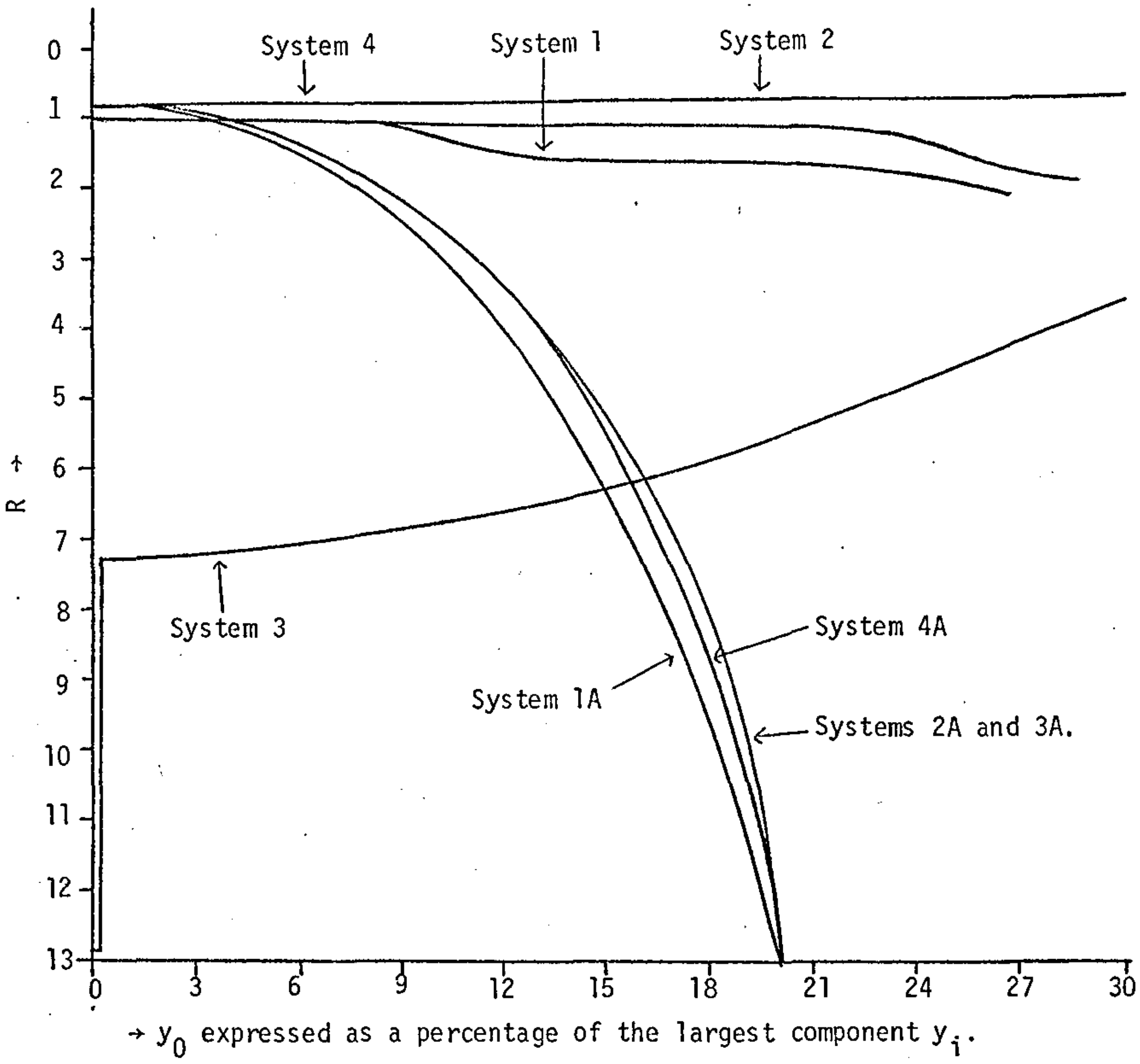


→ y_0 expressed as a percentage of the largest component y_i.

R is the reduction in tolerance to noise, when channel P replaces one which introduces no distortion or attenuation.

FIGURE 4.11

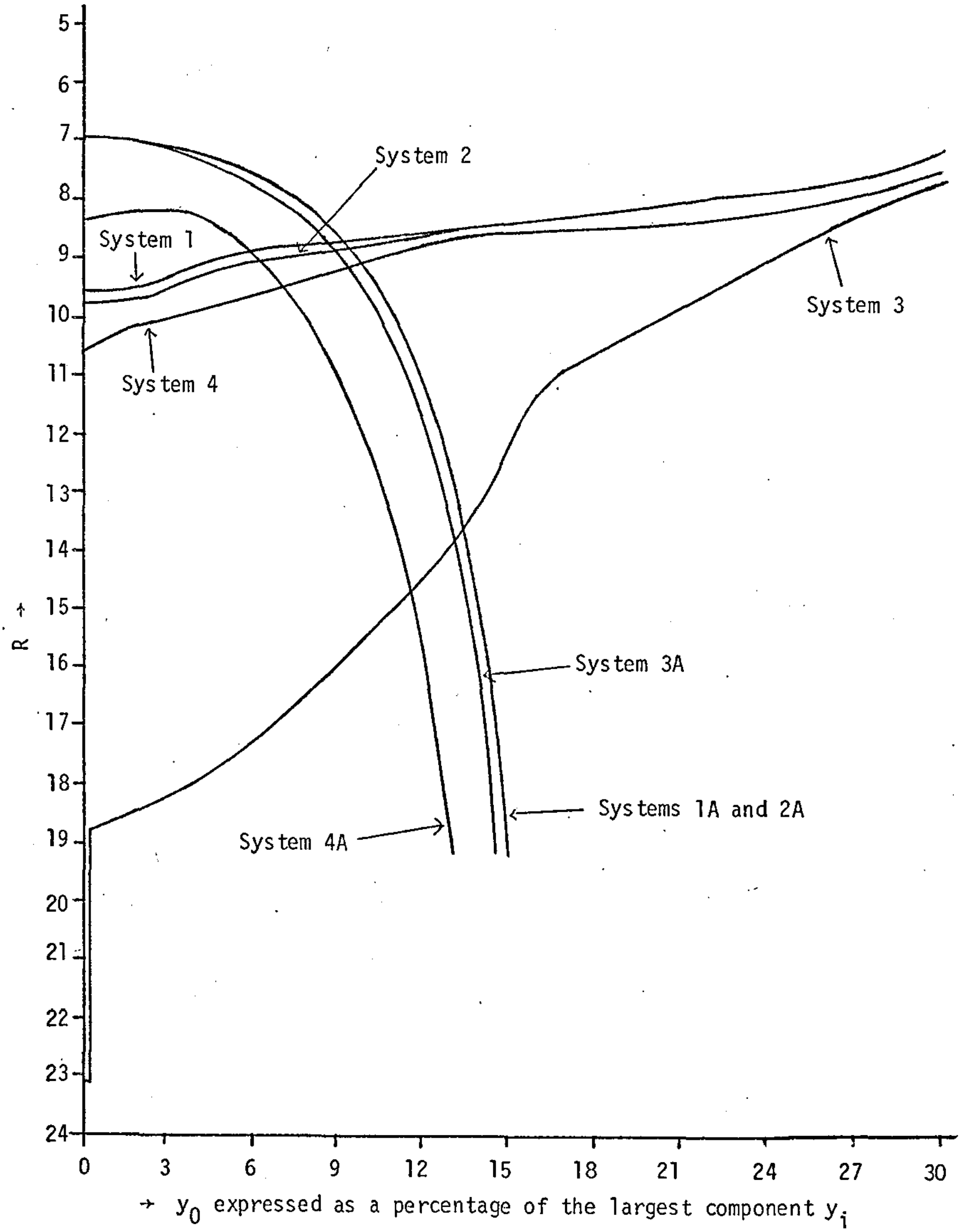
Variation in performance with y_0. Binary signals with channel P.



R is the reduction in tolerance to noise, when channel M replaces one which introduces no distortion or attenuation.

FIGURE 4.12

Variation in performance with y_0. Quaternary signals with channel M.



R is the reduction in tolerance to noise, when channel 0 replaces one introducing no distortion or attenuation.

FIGURE 4.13

Variation in performance with y_0. Quaternary signals with channel 0.

from channel A (in table 4.01), by the addition of the component y_0. System 1A ignores the first component of the channel vector. Hence, when used with channel M and $y_0 = 0$, System 1A is equivalent to System 1 used in conjunction with channel A. A similar relationship exists for Systems 2, 3 and 4, as shown in table 4.09(a).

From Section 3.10, it can be seen that using channels M-P with $y_0 = 0$, instead of channels A, D, G and J respectively, has the same effect on performance as would be obtained by reducing k by a factor of m. (The number of components of the vectors also being reduced by one). Hence System 1 with k = 16, N = 11 and channel M, is equivalent to System 1A with $k = \frac{16}{m}$, N = 10 and channel A. Similar relationships hold for System 2 and 4, and for System 1 with channels D, G and J. These are given in table 4.09(b).

It can be seen from table 4.09 that when $y_0 = 0$, using System 1A with M, N, O or P instead of System 1, has approximately the same effect on performance, as that of increasing the number of stored vectors. Now consider the distance between the points, at which the System 1 and the System 1A curves cross the vertical axes, in figures 4.10 - 4.13. At these points on the graph, $y_0 = 0$, so this distance approximately represents the difference in performance between System 1 with $k = \frac{16}{m}$, and System 1 with k = 16. This, of course, also applies for Systems 2 and 4. (From Section 3.10, it can be seen that setting $y_0 = 0$ has the same effect on each of the Systems 1, 2 and 4). It can however be seen from figures 4.10 - 4.13, that this property does not hold for System 3. With $y_0 = 0$, the improvement in performance offered by System 3A over System 3, can be much greater than that obtainable, by increasing the value of k

(a)

System	Channel
1A	M
1A	N
1A	O
1A	P
2A	M
2A	N
2A	O
2A	P
3A	M
3A	N
3A	O
3A	P
4A	M
4A	N
4A	O
4A	P

⟷

System	Channel
1	A
1	D
1	G
1	J
2	A
2	D
2	G
2	J
3	A
3	D
3	G
3	J
4	A
4	D
4	G
4	J

(b)

System	Channel	
1	M	
1	N	
1	O	
1	P	
2	M	k=16
2	N	N=11
2	O	
2	P	
4	M	
4	N	
4	O	
4	P	

⟷

System	Channel	
1	A	
1	D	
1	G	
1	J	
1	A	k=16/m
1	D	N=10
1	G	
1	J	
4	A	
4	D	
4	G	
4	J	

TABLE 4.09

Equivalent arrangements of detection processes and transmission channels, when $y_0 = 0$

by a factor of m. This improvement is particularly noticeable with Channel 0 and a quaternary signal, as can be seen from figure 4.13.

Clearly, if Systems 1-4 are used with a value of k, such that a reduction of k to k/m would cause a loss in performance, it is sometimes of advantage for the first channel component y_0, to be ignored. For such values of k, very small first channel components should always be ignored, by Systems 1-4. From figures 4.10 - 4.13, it may be assessed to some extent, just how small y_0 should be, for it to be ignored. Consider, for example, System 1 with k = 16, a four level signal and channel 0. The value of y_0 such that Systems 1 and 1A yield the same tolerance to noise, is given by the point at which the corresponding curves intersect, in figure 4.13. The intersection occurs when y_0 is about 9% of the largest component of the channel vector. Hence, for this case, the first channel component is best ignored, if it is less than 9% of the peak component.

Consider now any of the Systems 1-4, together with its modified version, in which the first channel component y_0 is ignored. It may be seen from figures 4.10 - 4.13, that the improvement in performance offered by the modified system, is greatest when y_0 is equal to zero. (Only non negative values of y_0 were considered for the tests). As y_0 increases, the performances of the original and modified systems become closer together, until the original system offers the best tolerance to noise. It can be seen, therefore, that the improvement in performance given by Systems 1A-4A, over Systems 1-4, is bounded above by the improvement given when $y_0 = 0$. For Systems 1, 2 and 4, the improvement given when $y_0 = 0$, is approximately that which would be obtained by increasing k by a factor of m. (See Section 3.10). Hence, for Systems 1, 2 and 4, the improvement offered by their modified versions, is bounded above by the improvement obtainable by increasing k by a factor of m. This bound is applicable for all non negative values of y_0.

One point that really stands out from figures 4.10 - 4.13, is that System 3 can suffer to a much greater extent that Systems 1, 2 and 4, from the effect of an extra zero at the start of the channel vector. Figure 4.13 shows that System 3 has a loss in tolerance to noise of about 13 db, over Systems 1, 2 and 4, when the first channel component is equal to zero. However, a very small increase in the value of y_0, improves the performance of System 3 by about 4 db.

It is clear from the above discussion that System 3 can sometimes give a very poor performance, if the first component of the channel vector is small. If a situation occurs, where such a small component cannot be ignored in the detection process, then it would appear that Systems 1, 2 and 4 are to be preferred to System 3.

A comparison of figures 4.10 - 4.13 suggests, that placing an extra small component at the start of the channel vector, has a more serious effect on the detection process with a quaternary signal, than it does if a binary signal is used. It seems likely that this effect will be even more pronounced, if the number of signal levels is increased beyond four. (This can be seen to be the case for Systems 1, 2 and 4, from Section 3.10).

CHAPTER 5

5.01 The Erratic Performance of Systems 1 and 2

From the simulation results given in tables 4.05 and 4.06, it would appear that Systems 1-4 each offer roughly the same tolerance to additive white Gaussian noise. It was found, however, that during some of the longer simulation tests described in Section 4.08, Systems 1 and 2 occasionally experienced a sudden drop in performance, i.e. a sudden and significant increase in the error rate. A close examination of the computer print-out for these tests, revealed that the reduction in performance was concurrent with some of the stored vectors becoming identical to each other, and their costs becoming almost the same. It will be seen from the following discussion, that Systems 1 and 2 can become locked in a state, in which several identical vectors are always present. Clearly, the number of possible data sequences that can be stored, is reduced if the detection process enters this state. The detectors are then effectively working with a lower number of stored vectors, and their performances may be reduced.

From Section 3.08, it can be seen that System 1 will become locked in a state where all of its stored vectors and costs are identical, if it should enter this state at any time. Similarly, System 2 can become locked in a state where the vectors divide into m groups, with the vectors and costs in a group being identical (m is the number of signal levels). The following analysis shows that, if two of the vectors stored by Systems 1 and 2 become the same, and their costs become close together, the performances of the detectors can be reduced. This phenomenon in which vectors become the same, and their costs become identical, or close together, is referred to as merging.

In order to simplify the analysis of the merging phenomenon, it will be assumed from here onward, that a two level data signal is used. Then the possible values of the data elements s_i are ±1. (The merging phenomenon has been examined in reference 49 which contains some of the work from this chapter).

5.02 Probability of Merging

Definition

Two vectors are said to be merged together, with separation ε, if their latest g components are the same, and their costs differ by an amount equal to ε. (g+1 is the number of components of the channel vector under consideration). If two vectors are said to be merged together, it is assumed that the separation is small.

Now consider the first cycle of the detection process, in which two vectors are present, which are merged with a separation less than some given amount ε. Let these two vectors be defined by

$$\underline{Q}_{j+1}(I) = [x_{j-N+2}(I), \ \ldots\ldots, \ x_{j-g+1}(I), \ x_{j-g+2}, \ x_{j-g+3}, \ldots\ldots, \ x_{j+1}]$$

and

$$\underline{Q}_{j+1}(K) = [x_{j-N+2}(K), \ \ldots\ldots, \ x_{j-g+1}(K), \ x_{j-g+2}, \ x_{j-g+3}, \ \ldots\ldots, x_{j+1}]$$

so that the latest g components are the same for both vectors. As before, x_i is a possible value of the data element s_i. $x_i(I)$ is the possible value of s_i which is present in the Ith stored vector $\underline{Q}_{j+1}(I)$. The only two vectors at the end of the previous cycle of the process, which can lead to $\underline{Q}_{j+1}(I)$ are

$$[\pm 1,\ x_{j-N+2}(I),\ \ldots.,\ x_{j-g+1}(I),\ x_{j-g+2},\ x_{j-g+3},\ \ldots.,\ x_j]$$

(Note that a binary signal is assumed throughout this chapter, so the possible data element values are ±1. Similarly, the only two vectors that can be extended to form $\underline{Q}_{j+1}(K)$ are

$$[\pm 1,\ x_{j-N+2}(K),\ \ldots.,\ x_{j-g+1}(K),\ x_{j-g+2},\ x_{j-g+3},\ \ldots,\ x_j]$$

Hence it is clear that two vectors with their latest g-1 components in common, must be present at the end of one cycle of the detection process, if two merged vectors are to appear during the following cycle.

A situation will now be considered in which two vectors, with their latest g-1 elements in common, are present at the end of some cycle of the process. The transmitted data signal is assumed to be a binary one, so these two vectors will be extended to four, in the following cycle. (See Section 3.02). It will then be demonstrated that there is only a small probability, of two of these extended vectors being merged together.

Now define the vectors $\underline{Q}_j(I)$ and $\underline{Q}_j(K)$ so that

$$\underline{Q}_j(I) = [x_{j-N+1}(I),\ \ldots,\ x_{j-g+1}(I),\ x_{j-g+2},\ x_{j-g+3},\ \ldots,\ x_j] \tag{5.01}$$

and

$$\underline{Q}_j(K) = [x_{j-N+1}(K),\ \ldots,\ x_{j-g+1}(K),\ x_{j-g+2},\ x_{j-g+3},\ \ldots,\ x_j] \tag{5.02}$$

(Note that these vectors have their latest g-1 elements in common). Let the costs for the vectors be $u_j(I)$ and $u_j(K)$ respectively. Also let the vectors $\underline{Q}_j(I)$ and $\underline{Q}_j(K)$, be such that they are not merged with a separation $\leq \varepsilon$, for some given value ε. Then, from the definition of merged vectors given above, at least one of the following conditions must be satisfied:

$$\text{i)} \quad x_{j-g+1}(I) \neq x_{j-g+1}(K) \qquad (5.03)$$

$$\text{ii)} \quad |u_j(I) - u_j(K)| > \varepsilon \,. \qquad (5.04)$$

Let $\underline{Q}_j(I)$ and $\underline{Q}_j(K)$ be two vectors which are present, at the end of the j+1 st. cycle of a System 1 or a System 2 detection process. Then, in the j+2 nd. cycle, these two vectors will be extended to the four vectors given by

$$\underline{T}_{j+1}\ (I,\ x_{j+1}) = [\underline{Q}_j(I),\ x_{j+1}]\,. \qquad (5.05)$$

and

$$\underline{T}_{j+1}\ (K,\ x_{j+1}) = [\underline{Q}_j(K),\ x_{j+1}] \qquad (5.06)$$

for $x_{j+1} = \pm 1$. (See Section 3.02). The costs for these vectors are given by

$$v_{j+1}\ (I,\ x_{j+1}) = u_j(I) + \{\underline{Y}.[\underline{T}_{j+1}\ (I,\ x_{j+1})]_{g+1} - r_{j+1}\}^2 \quad (5.07)$$

and

$$v_{j+1}\ (K,\ x_{j+1}) = u_j(K) + \{\underline{Y}.[\underline{T}_{j+1}\ (K,\ x_{j+1})]_{g+1} - r_{j+1}\}^2 \quad (5.08)$$

where $\underline{Y}$ is the reverse of the channel vector, and r_{j+1} is the j+2 nd. received signal sample. $[\underline{T}_{j+1}\ (I,\ x_{j+1})]_{g+1}$ is the vector formed from the latest g+1 components of $\underline{T}_{j+1}$ (I, x_{j+1}), so it can be seen that only these components are used for the evaluation of the cost v_{j+1} (I, x_{j+1}). Let w_{j+1} be the noise sample which contributes to r_{j+1}, and let

$$R_{j+1} = r_{j+1} - w_{j+1} \qquad (5.09)$$

Then R_{j+1} is the value of r_{j+1}, assuming that there is no noise in the system. But

$$\underline{Y}.[\underline{T}_{j+1} (I, x_{j+1})]_{g+1}$$

is defined to be the scalar product of the vector

$$(y_g, y_{g-1}, \ldots\ldots, y_0)$$

and the latest g+1 components of the vector

$$\underline{T}_{j+1} (I, x_{j+1})$$

(see Section 3.02). Hence, using equations 5.01 and 5.05,

$$\underline{Y}.[\underline{T}_{j+1} (I, x_{j+1})]_{g+1}$$

$$= y_g\, x_{j-g+1}(I) + y_{g-1}\, x_{j-g+2} + \ldots\ldots + y_0\, x_{j+1}$$

or

$$\underline{Y}.[\underline{T}_{j+1} (I, x_{j+1})]_{g+1} = y_g\, x_{j-g+1} (I) + \alpha \qquad (5.10)$$

where

$$\alpha = y_{g-1}\, x_{j-g+2} + y_{g-2}\, x_{j-g+3} + \ldots\ldots + y_0\, x_{j+1} \qquad (5.11)$$

Similarly,

$$\underline{Y}.[\underline{T}_{j+1}(K, x_{j+1})]_{g+1} = y_g\ x_{j-g+1}(K) + \alpha \qquad (5.12)$$

Now, substituting equations 5.10 and 5.12 into equations 5.07 and 5.08 gives

$$v_{j+1}(I, x_{j+1}) = u_j(I) + \{y_g\ x_{j-g+1}(I) + \alpha - r_{j+1}\}^2$$

and

$$v_{j+1}(K, x_{j+1}) = u_j(K) + \{y_g\ x_{j-g+1}(K) + \alpha - r_{j+1}\}^2.$$

But $r_{j+1} = R_{j+1} + w_{j+1}$
(see equation 5.09), therefore

$$v_{j+1}(I, x_{j+1}) = u_j(I) + \{y_g\ x_{j-g+1}(I) + \alpha - R_{j+1} - w_{j+1}\}^2$$

and

$$v_{j+1}(K, x_{j+1}) = u_j(K) + \{y_g\ x_{j-g+1}(K) + \alpha - R_{j+1} - w_{j+1}\}^2$$

Subtracting the second equation from the first gives

$$v_{j+1}(I, x_{j+1}) - v_{j+1}(K, x_{j+1}) = u_j(I) - u_j(K)$$

$$+ \{y_g\ x_{j-g+1}(I) + \alpha - R_{j+1} - w_{j+1}\}^2 - \{y_g\ x_{j-g+1}(K) + \alpha - R_{j+1} - w_{j+1}\}^2$$

Therefore

$$v_{j+1}(I, x_{j+1}) - v_{j+1}(K, x_{j+1})$$

$$= u_j(I) - u_j(K)$$

$$+ \{y_g\ [x_{j-g+1}(I) + x_{j-g+1}(K)] + 2\alpha - 2R_{j+1} - 2w_{j+1}\}$$

$$\times\ y_g\ \{x_{j-g+1}(I) - x_{j-g+1}(K)\} \qquad (5.13)$$

First consider a case where

$$x_{j-g+1}(I) = x_{j-g+1}(K).$$

Then equation 5.13 reduces to

$$v_{j+1}(I, x_{j+1}) - v_{j+1}(K, x_{j+1}) = u_j(I) - u_j(K).$$

From the inequality 5.04,

$$|u_j(I) - u_j(K)| > \varepsilon$$

$$\therefore \quad |v_{j+1}(I, x_{j+1}) - v_{j+1}(K, x_{j+1})| > \varepsilon$$

and the vectors $\underline{T}_{j+1}(I, x_{j+1})$ and $\underline{T}_{j+1}(K, x_{j+1})$ cannot be merged with a separation which is $\leq \varepsilon$.

Now consider a case where

$$x_{j+g+1}(I) \neq x_{j+g+1}(K).$$

Then, from equation 5.13,

$$v_{j+1}(I, x_{j+1}) - v_{j+1}(K, x_{j+1}) = u_j(I) - u_j(K)$$

$$+ (c_1 - 2w_{j+1})\, c_2$$

where

$$c_1 = y_g\, [x_{j-g+1}(I) + x_{j-g+1}(K)] + 2\alpha - 2R_{j+1}$$

and

$$c_2 = y_g\, \{x_{j-g+1}(I) - x_{j-g+1}(K)\} \qquad (5.14)$$

Hence

$$v_{j+1}(I, x_{j+1}) - v_{j+1}(K, x_{j+1}) = u_j(I) - u_j(K) + c_1c_2 - 2c_2\, w_{j+1},$$

and the condition:

$$|v_{j+1}(I, x_{j+1}) - v_{j+1}(K, x_{j+1})| \leq \varepsilon$$

is satisfied only if $2c_2\, w_{j+1}$ lies in an interval of width 2ε i.e. w_{j+1} must lie in an interval of width

$$\frac{\varepsilon}{|c_2|} .$$

It follows, therefore, that the vectors

$$\underline{T}_{j+1}(I, x_{j+1}) \quad \text{and} \quad \underline{T}_{j+1}(K, x_{j+1})$$

can be merged with a separation $\leq \varepsilon$, only if w_{j+1} lies in some given interval of width $\varepsilon/|c_2|$.

In the assumed model of a data transmission system (see Section 1.02), w_{j+1} is a normally distributed random variable with zero mean, and variance denoted σ^2. Hence the probability density function for w_{j+1} is given by

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-x^2}{2\sigma^2}\right)$$

A rough sketch of f(x), plotted against x, is given in Figure 5.01. The probability of w_{j+1} lying in any interval (a, b) is given by

$$\Pr\,(a < w_{j+1} < b) = \int_a^b f(x)\, dx.$$

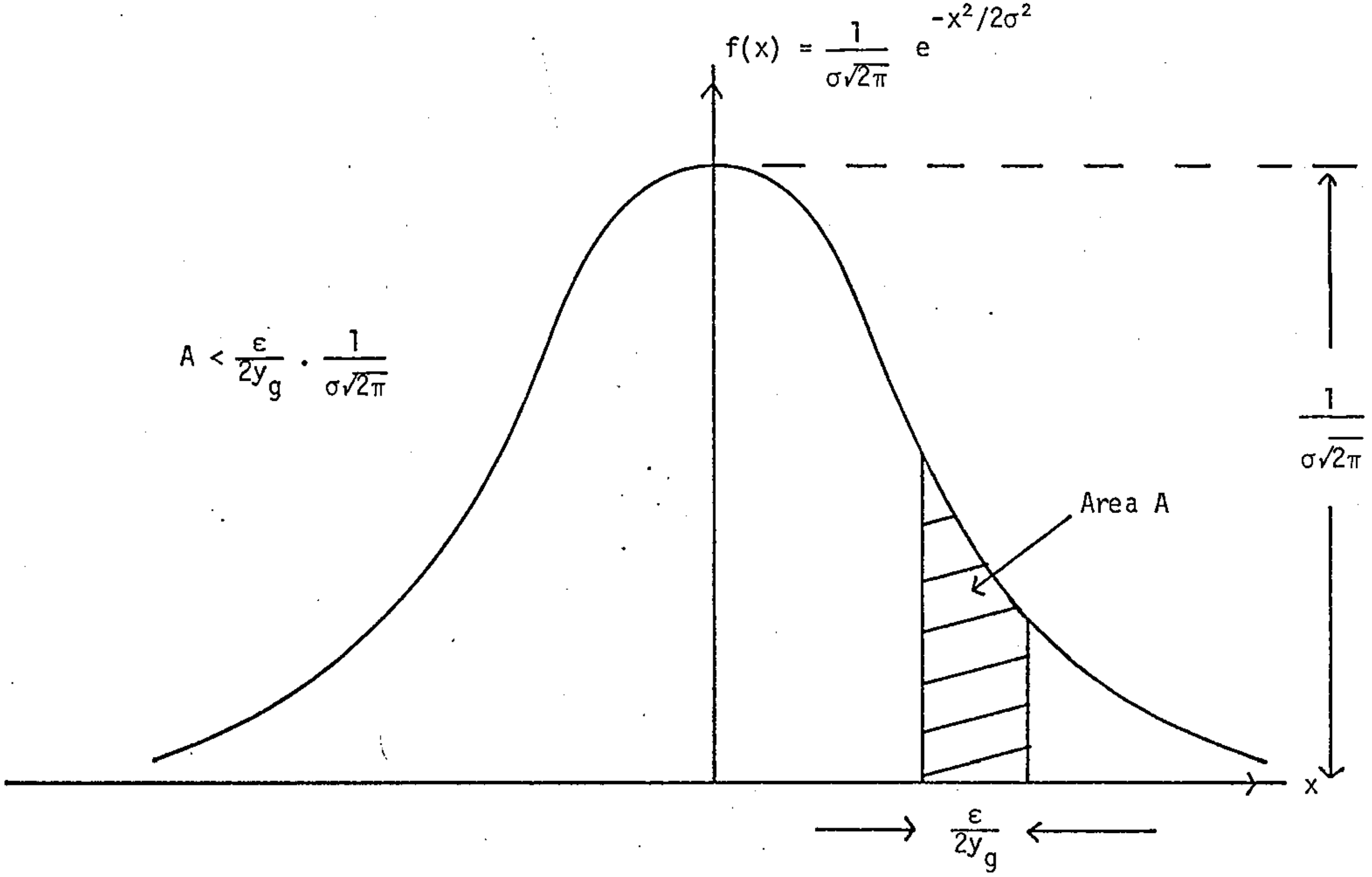

$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/2\sigma^2}$
$A < \frac{\varepsilon}{2y_g} \cdot \frac{1}{\sigma\sqrt{2\pi}}$
Area A
$\frac{1}{\sigma\sqrt{2\pi}}$
x
$\frac{\varepsilon}{2y_g}$

FIGURE 5.01

Probability density function for a random variable with zero mean and variance σ^2.

Hence

$$\Pr\ (a < w_{j+1} < b) \leq \int_a^b f_{max}\, dx$$

$$\leq (b-a)\, f_{max}$$

where f_{max} is the maximum value of $f(x)$.

But

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-x^2}{2\sigma^2}\right)$$

so $f_{max} = (\sigma\sqrt{2\pi})^{-1}$.

Therefore

$$\Pr\ (a < w_{j+1} < b) \leq \frac{b-a}{\sigma\sqrt{2\pi}}\ . \qquad (5.15)$$

It has been shown above, that the vectors $\underline{T}_{j+1}(I, x_{j+1})$ and $\underline{T}_{j+1}(K, x_{j+1})$ can be merged with separation $\leq \varepsilon$, only if w_{j+1} lies in some given interval of width $\varepsilon/|c_2|$,

i.e.

$$P \leq \text{Prob. } (w_{j+1} \text{ lies in an interval of width } \varepsilon/|c_2|)$$

where P is the probability that $\underline{T}_{j+1}(I, x_{j+1})$ and $\underline{T}_{j+1}(K, x_{j+1})$ are merged with a separation $\leq \varepsilon$. Now using the inequality 5.15 gives

$$P \leq \frac{\varepsilon}{|c_2|\ \sigma\sqrt{2\pi}}\ . \qquad (5.16)$$

where c_2 is given by equation 5.14. (Note that this bound for P has been derived for a case where $x_{j-g+1}(I) \neq x_{j-g+1}(K)$).

This bound for P will now be applied, to one of the situations covered by the simulation tests, described in Chapter 4. A binary signal is assumed throughout this chapter, so that the possible data element values are ±1. Therefore, assuming that

$$x_{j-g+1}(I) \neq x_{j-g+1}(K),$$

equation 5.14 gives

$$|c_2| = 2|y_g|.$$

Consider now, the simulation tests carried out on channel E, at an error rate of 0.004, and a two level signal. Then $y_g = 0.167$ (see Table 4.01) and the appropriate value of σ is 0.18, for System 1 with k=4. The inequality 5.16, therefore becomes

$$P \leq \frac{\varepsilon}{2 \times 0.167 \times 0.18 \times \sqrt{2\pi}}$$

or

$$P \leq 6.6\,\varepsilon$$

In simulation tests it was found that a noticeable drop in performance, with Systems 1 and 2, was accompanied by vectors merged with a separation ε of 10^{-3} or less. Hence the probability P of the vectors $\underline{T}_{j+1}(I, x_{j+1})$ and $\underline{T}_{j+1}(K, x_{j+1})$ being merged with a separation small enough to reduce performance, is $\leq 6.6 \times 10^{-3}$. It should be noted that the inequality 5.15, gives a tight bound only if a and b are close to zero. In general,

$$Pr\,(a < w_{j+1} < b) \ll \frac{b-a}{\sigma\sqrt{2\pi}} \quad \text{and}$$

$$P \ll 6.6 \times 10^{-3}$$

(i.e. P is much less than 6.6×10^{-3}).

Summary

From the above analysis, it is clear that two vectors of the form $\underline{Q}_j(I)$ and $\underline{Q}_j(K)$, must be present at the end of the j+1st. cycle of the process, if merged vectors are to appear in the following cycle. If such vectors are present at the end of the j+1st. cycle, there is then a small probability of vectors appearing in the j+2nd. cycle, which are merged with a significantly small separation. If two merged vectors do appear in the set of expanded vectors, in some cycle of the detection process, these vectors may not be selected by the appropriate decision rule. If they are not selected, the following cycle of the process will begin without any merged vectors.

5.03 Non-Merged Vectors Stemming from Merged Vectors

As before, the System 1 and System 2 detection processes will be considered. Let $\underline{Q}_j(I)$ and $\underline{Q}_j(K)$ be two vectors present at the end of the j+1st. cycle, which are merged with some small separation ε. Then, from the definition of merged vectors, $\underline{Q}_j(I)$ and $\underline{Q}_j(K)$ must have their latest g elements in common. (See Section 5.02).

Let

$$\underline{Q}_j(I) = [x_{j-N+1}(I), x_{j-N+2}(I), \ldots, x_{j-g}(I), x_{j-g+1}, x_{j-g+2}, \ldots, x_j]$$

and

$$\underline{Q}_j(K) = [x_{j-N+1}(K), x_{j-N+2}(K), \ldots, x_{j-g}(K), x_{j-g+1}, x_{j-g+2}, \ldots, x_j]$$

Let the costs for these vectors be $u_j(I)$ and $u_j(K)$ respectively, and let

$$u_j(K) = u_j(I) + \varepsilon \tag{5.17}$$

In the j+2nd. cycle of the process, $\underline{Q}_j(I)$ and $\underline{Q}_j(K)$ are extended to the four vectors given by

$$\underline{T}_{j+1}\ (I,\ x_{j+1}) = [\underline{Q}_j(I),\ x_{j+1}] \tag{5.18}$$

and

$$\underline{T}_{j+1}\ (K,\ x_{j+1}) = [\underline{Q}_j(K),\ x_{j+1}] \tag{5.19}$$

where x_{j+1} may take the values ±1. (Assuming, as before, that a binary data signal is used). The costs for these four vectors are given by

$$v_{j+1}(I,\ x_{j+1}) = u_j(I) + \{\underline{Y}.[\underline{T}_{j+1}(I,\ x_{j+1})]_{g+1} - r_j\}^2 \tag{5.20}$$

and

$$v_{j+1}(K,\ x_{j+1}) = u_j(K) + \{\underline{Y}.[\underline{T}_{j+1}(K,\ x_{j+1})]_{g+1} - r_j\}^2 \tag{5.21}$$

where $\underline{Y}$ is the reverse of the channel vector, and r_j is the j+1st. received signal sample. (See Section 3.02). The four vectors of the forms $\underline{T}_{j+1}\ (I,\ x_{j+1})$ and $\underline{T}_{j+1}\ (K,\ x_{j+1})$ may be said to have stemmed from $\underline{Q}_j(I)$ and $\underline{Q}_j(K)$.

From the definitions of $\underline{Q}_j(I)$ and $\underline{Q}_j(K)$, these two vectors have their latest g elements in common. Hence, for a given value of x_{j+1}, the two vectors $\underline{T}_{j+1}(I,\ x_{j+1})$ and $\underline{T}_{j+1}(K,\ x_{j+1})$, have their latest g+1 elements in common. (See equations 5.18 and 5.19)

Hence

$$\underline{Y}.[\underline{T}_{j+1}(I, x_{j+1})]_{g+1} = \underline{Y}.[\underline{T}_{j+1}(K, x_{j+1})]_{g+1}$$

as in general $\underline{Y}.[\underline{v}]_{g+1}$ is defined to be the scalar product of $\underline{Y}$, and the g+1 components of $\underline{v}$, which are furthest to the right. This definition applies to any vector $\underline{v}$ with g+1, or more, components. Hence, from equations 5.20 and 5.21,

$$v_{j+1}(I, x_{j+1}) - v_{j+1}(K, x_{j+1}) = u_j(I) - u_j(K)$$

But $|u_j(I) - u_j(K)| = \varepsilon$

as the vectors $\underline{Q}_j(I)$ and $\underline{Q}_j(K)$ are defined to be merged with a separation ε. Therefore

$$|v_{j+1}(I, x_{j+1}) - v_{j+1}(K, x_{j+1})| = \varepsilon.$$

It has been seen above that the vectors $\underline{T}_{j+1}(I, x_{j+1})$ and $\underline{T}_{j+1}(K, x_{j+1})$ have their latest g elements in common, for a given value of x_{j+1}. Hence $\underline{T}_{j+1}(I, -1)$ and $\underline{T}_{j+1}(K, -1)$ are merged with separation ε. Also, $\underline{T}_{j+1}(I, 1)$ and $\underline{T}_{j+1}(K, 1)$ are merged with a separation ε. (Note that ±1 are the allowable values for x_{j+1}).

At the end of the j+1 st. cycle of the detection process, it was assumed that there were two vectors present, which were merged with some small separation ε. The above analysis, shows that this situation leads to one, in which two pairs of merged vectors are present during the following cycle (assuming a binary data signal). It is therefore clear that the number of merged vectors present in Systems 1 and 2, can increase from one cycle to the next. It is also evident that non-merged vectors cannot stem from a pair of merged vectors.

5.04 Selection of Merged Vectors

The recommended starting up procedure, for Systems 1 and 2 (and Systems 3 and 4) is given in Section 3.08. With this procedure, the initial set of k vectors are such that, one of them has zero cost and the others have infinite costs. This ensures that a distinct set of vectors will be present in the process, after a few data elements have been detected. (See the proof of theorem 3.02, Section 3.10). Then, in the following cycles of the detection process, there is a small probability that the set of expanded vectors, will contain a pair of vectors which are merged with a small separation. In each cycle of the process, the set of expanded vectors will number 2k, and half of these vectors will be selected according to the appropriate decision rule. (Note that the number of signal levels is assumed to be two).

In any cycle of a System 1 or a System 2 detection process, in which the set of expanded vectors contains a merged pair, this pair may or may not be retained for the following cycle. If the merged pair of vectors is selected, by the appropriate decision rule, there will be two merged pairs of vectors available for selection in the next cycle of the process. (See Section 5.03). It is then possible for the number of pairs of merged vectors to increase after each cycle.

Now assume that the number k, of vectors stored at the start of each cycle, is even. Then it is clearly possible for all of the vectors stored by Systems 1 and 2, to consist of pairs of merged vectors. It will be demonstrated that there is a high probability, of this situation being preserved from one cycle to the next.

Theorem 5.01

Let the number k, of vectors stored at the beginning of each cycle of a System 1 or a System 2 detection process, be even. Let the k vectors stored at the end of some cycle of the process, consist of pairs of vectors which are merged together with a separation $\leq \varepsilon$, for some small number ε. Vectors which are not in the same pair are assumed not to be merged. Let the k vectors be denoted $\underline{Q}_j(I)$, with costs $u_j(I)$ for

$$I = 1, 2, \ldots\ldots, k.$$

Also let these vectors be such that, any two of them which form a merged pair, have their latest M components in common, where

$$g \leq M \leq N.$$

Note that, by definition, merged vectors must have their latest g components, (i.e. the g components furthest to the right), in common. Let the costs for the extended vectors:

$$[\underline{Q}_j(I), x_{j+1}],$$

in the following cycle, be denoted $v_{j+1}(I, x_{j+1})$ and assume that

$$|v_{j+1}(I, -1) - v_{j+1}(J,-1)| > \varepsilon \tag{5.22}$$

and

$$|v_{j+1}(I, 1) - v_{j+1}(J, 1)| > \varepsilon \tag{5.23}$$

except when I and J are such that $\underline{Q}_j(I)$ and $\underline{Q}_j(J)$ are merged with a separation $\leq \varepsilon$. Then the k vectors selected by the decision rule for System 1, will consist of $\frac{1}{2}$k pairs of vectors, with the vectors in each pair being merged with a separation $\leq \varepsilon$. This also applies to System 2 if k is a multiple of four.

Furthermore, any two of these selected vectors which form a merged pair, will have their latest M+1 components in common (i.e. the M+1 components furthest to the right will be the same in both vectors).

Proof

a) First consider the vectors selected by System 1. The decision rule for System 1 ensures that the k vectors with smallest costs will be selected from the set of 2k extended vectors, of the form

$$[\underline{Q}_j(I), x_{j+1}]$$

Let I_1 and I_2 be such that the vectors $\underline{Q}_j(I_1)$ and $\underline{Q}_j(I_2)$ form one of the merged pairs, and the corresponding costs, $u_j(I_1)$ and $u_j(I_2)$, are such that

$$0 \leq u_j(I_2) - u_j(I_1) \leq \varepsilon.$$

From Section 5.03, it can be seen that

$$v_{j+1}(I_2, x_{j+1}) - v_{j+1}(I_1, x_{j+1}) = u_j(I_2) - u_j(I_1)$$

where $v_{j+1}(I, x_{j+1})$ is the cost associated with the vector $[\underline{Q}_j(I), x_{j+1}]$.

From the inequalities 5.22 and 5.23, it can be seen that $v_{j+1}(I_1, x_{j+1})$ is the only cost within an amount ε of $v_{j+1}(I_2, x_{j+1})$.

Similarly $v_{j+1}(I_2, x_{j+1})$ is the only cost within an amount ε of $v_{j+1}(I_1, x_{j+1})$. Hence it is clear that, whichever of these costs is selected first, the other one will be the next to be chosen, by the decision rule for System 1. The k costs to be selected will be selected in pairs of the form

$$v_{j+1}(I, x_{j+1}), \quad v_{j+1}(J, x_{j+1})$$

where I and J are such that the vectors $\underline{Q}_j(I)$ and $\underline{Q}_j(J)$ are merged with a separation $\leq \varepsilon$. Hence the vectors selected by decision rule one, will be selected in pairs of the form

$$[\underline{Q}_j(I), x_{j+1}] \, , \quad [\underline{Q}_j(J), x_{j+1}]$$

with the vectors in each pair being merged with a separation $\lesssim \varepsilon$. The k selected vectors will therefore consist of $\frac{1}{2}k$ such pairs. $\underline{Q}_j(I)$ and $\underline{Q}_j(J)$ are such that the latest M components are common to the two vectors. Hence the latest M+1 components will be in common, for any of the vectors making up a merged pair of the form

$$[\underline{Q}_j(I), x_{j+1}] \, , \qquad [\underline{Q}_j(J), x_{j+1}]$$

b) Now assume that k is a multiple of four, and consider the vectors selected by the decision rule for System 2. Decision rule 2 considers the set of 2k vectors

$$\{[\underline{Q}_j(I), x_{j+1}]\}$$

as being divided into two separate groups, according to their value of x_{j+1}. The ½k vectors with smallest costs are selected, from those which have $x_{j+1} = -1$. This is repeated for the vectors having $x_{j+1} = 1$. As with the proof for System 1,

$$v_{j+1}(I_1, x_{j+1}) \quad \text{and} \quad v_{j+1}(I_2, x_{j+1})$$

are the costs which are closest to each other, if I_1 and I_2 are such that $\underline{Q}_j(I_1)$ and $\underline{Q}_j(I_2)$ form one of the merged pairs. Hence the two costs

$$v_{j+1}(I_1, -1) \text{ and } v_{j+1}(I_2, -1)$$

will either both be selected, or neither of them will be selected. The vectors selected for which $x_{j+1} = -1$ will therefore be chosen in pairs of the form

$$[\underline{Q}_j(I), -1)], [\underline{Q}_j(J), -1]$$

where $\underline{Q}_j(I)$ and $\underline{Q}_j(J)$ are merged with a separation $\leq \varepsilon$. Hence the ½k selected vectors with $x_{j+1} = -1$, will consist of ¼k pairs of vectors, with the vectors in each pair being merged with a separation $\leq \varepsilon$. Clearly, the same applies to the selected vectors which have $x_{j+1} = 1$. As with System 1, the latest M+1 components will be in common, for any two vectors forming a merged pair such as

$$[\underline{Q}_j(I), x_{j+1}], \quad [\underline{Q}_j(J), x_{j+1}]$$

End of proof of theorem 5.01.

Corrolary to theorem 5.01

The assumptions given by inequalities 5.22 and 5.23 have been found, from simulation tests, to be nearly always valid if $\varepsilon <$ about 10^{-3}. Hence theorem 5.01 may be applied to any situation, where the vectors stored by System 1 or System 2 at the end of some cycle, consist of pairs of vectors merged with a small separation. There is then a high probability that this situation will be maintained in the next cycle of the process. It is therefore possible that System 1 with k even, and System 2 with k being a multiple of four, can become locked in a state where the stored vectors are formed into merged pairs. These systems have been observed to remain in this state for tens of thousands of cycles, during simulation tests.

Definition

The failure mode for Systems 1 and 2, with some given value of ε, is defined to be a state in which all of the stored vectors are formed into merged pairs, with a separation $\leq \varepsilon$.

The term, "failure mode", is really appropriate only if the value of ε is small. For small values of ε, it has been seen that there is only a small probability of the systems escaping from this mode, in any particular cycle. The definition only covers a situation in which a binary data signal is used, and the number of vectors stored at the start of each cycle is even.

Now consider a situation in which System 1 or System 2 enters the failure mode, during some given cycle j. Then the k vectors stored at the start of cycle j+1, will form $\frac{1}{2}$k pairs of merged vectors.

From the definition of merged vectors, any two vectors forming a merged pair at the start of cycle j+1, must have their latest g components in common. Also, from theorem 5.01, it can be seen that any two vectors forming a merged pair at the start of cycle j+2, will have at least their latest g+1 elements in common. The number of elements in common, for two vectors forming a merged pair, must increase after each cycle until this number reaches N. The system will then be locked in a state in which its stored vectors are arranged in pairs, with the two vectors in any pair being identical. Then only $\frac{1}{2}$k different data sequences can be stored at the start of each detection cycle, and the process is effectively working with k reduced to half. The detectors then require more storage, and perform more calculations, than required for a given performance. Clearly, the failure mode is an undesirable state of operation for Systems 1 and 2.

From the above discussion, it seems advisable that System 1 be used with an odd value of k, and System 2 be used with a value which is not a multiple of four. Theorem 5.01 is not then applicable, and Systems 1 and 2 should not become locked in the failure mode.

5.05 Probability of System 1 Eventually Entering the Failure Mode, Given that a Particular Pair of Merged Vectors are Present at Some Stage

From Section 5.04, it can be seen that System 1 may eventually enter the failure mode, if a merged pair of vectors is formed during some cycle of the process. It is, of course, necessary that some merged pairs of vectors are selected by the decision rule in each cycle, if the failure mode is to occur.

Now consider a situation where two merged vectors, $\underline{Q}_j(I_1)$ and $\underline{Q}_j(I_2)$ are formed, with some small separation ε, during some cycle of a System 1 detection process. This pair is assumed to be the only pair of merged vectors present, during the j+1st. cycle. It can be seen from Section 5.02, that there is only a small probability of merged vectors appearing in the following cycle, which have not stemmed from $\underline{Q}_j(I_1)$ and $\underline{Q}_j(I_2)$. This probability will now be assumed to be negligibly small. In the j+2nd. cycle of the detection process, two pairs of merged vectors (separation ε) will stem from $\underline{Q}_j(I_1)$ and $\underline{Q}_j(I_2)$. (See Section 5.03). Hence the decision rule for System 1, may select zero, one or two pairs of merged vectors in the j+2nd. cycle.

A situation will now be examined in detail, in which System 1 is used with a binary signal, and four vectors stored at the start of each cycle, (i.e. m=2 and k=4). It will be assumed that two of the four vectors stored, at the end of the j+1st. cycle of the process, are merged together with separation zero. The remaining two vectors are assumed not to be merged with each other, or with the first pair. The effect of the selection procedure, on the occurrence of the failure mode, will then be assessed.

The four vectors present at the end of the j+1st. cycle of the process, will be extended to form eight vectors in the j+2nd. cycle. From Section 5.03, it can be seen that four of the vectors will be formed into two merged pairs, with separation zero. Only four of the eight vectors will be selected, and retained for use in the j+3rd. cycle.

From this point onward in the thesis, it will be assumed that the k stored vectors of the form $\underline{Q}_j(I)$, for System 1, are denoted in such a way that

$$u_j(I) \leq u_j(K) \quad \text{if} \quad I < K$$

where $u_j(I)$ is the cost associated with the vector $\underline{Q}_j(I)$.

The four vectors stored at the end of cycle j+1 are denoted $\underline{Q}_j(I)$, with costs $u_j(I)$, for I = 1, 2, 3, 4. In the j+2nd. cycle, the eight extended vectors are given by

$$\underline{T}_{j+1}(I, x_{j+1}) = [\underline{Q}_j(I), x_{j+1}]$$

with costs

$$v_{j+1}(I, x_{j+1})$$

for

$$I = 1, 2, 3, 4 \quad \text{and} \quad x_{j+1} = \pm 1.$$

The vectors $\underline{Q}_j(I_1)$ and $\underline{Q}_j(I_2)$ are merged with zero separation, so

$$\underline{T}_{j+1}(I_1, x_{j+1}) \quad \text{and} \quad \underline{T}_{j+1}(I_2, x_{j+1})$$

will also be merged with zero separation, for $x_{j+1} = \pm 1$. (See Section 5.03). Hence

$$v_{j+1}\ (I_1, -1) = v_{j+1}(I_2, -1) \quad \text{and}$$

$$v_{j+1}\ (I_1,\ 1) = v_{j+1}(I_2,\ 1).$$

Four of the eight vectors, of the form $\underline{T}_{j+1}(I, x_{j+1})$, must be selected by the decision rule for System 1. The latest N components of the selected vectors, then form the vectors

$$\underline{Q}_{j+1}(1), \underline{Q}_{j+1}(2), \underline{Q}_{j+1}(3), \underline{Q}_{j+1}(4)$$

which are arranged in ascending order of their costs. It can be seen that this set of four vectors may contain zero, one or two pairs of vectors, which are merged with separation zero.

Various modes of operation will now be defined for System 1, corresponding to the number of pairs of merged vectors, which are present in any given cycle. These modes are:

Mode R: The recovery mode. In this mode, the system has no merged vectors.

Mode I: In this mode, the vectors $\underline{Q}_j(I)$ and $\underline{Q}_j(I+1)$ are merged with zero separation. I may take on the values 1, 2 and 3, for the situation being considered.

Mode F: The failure mode. In this mode, vectors $\underline{Q}_j(1)$, $\underline{Q}_j(2)$ and $\underline{Q}_j(3)$, $\underline{Q}_j(4)$ form two pairs of vectors, which are merged with zero separation.

Clearly, if the detection process is in mode 1, 2 or 3, at the end of some cycle, it may change to any of the modes 1, 2, 3, recovery or failure, at the end of the following cycle. From Section 5.02, it is clear that there is only a small probability, that the system will go from a state of no merged vectors, to one with a pair of merged vectors, in any given cycle. Hence, once the process reaches the recovery mode, it will tend to remain in this mode. From Section 5.04, it can be seen that there is only a small probability of System 1 leaving the failure mode, in any cycle.

The transitions of the System 1 detection process, from one mode to another, may be described in terms of the trellis diagram of Figure 5.02. If the process enters one of the modes: 1, 2, or 3, in any cycle, it may then wander between these modes in the following cycles, until it settles at either mode R or mode F.

Let p_{IJ} be the probability of transition from mode I, in one cycle, to mode J in the following cycle. I may represent modes 1, 2 or 3 and J may represent any of the modes. The probabilities p_{IJ} are called transition probabilities. They are assumed to be constant from one cycle to the next.

Let $P_I(T)$ be the probability of the process being in mode I at the end of cycle T, where I may take the values 1, 2, 3, and T may take the values j, j+1, j+2, ...
(j is the first cycle which ends with a pair of merged vectors).
Let $P_I{}^*(T)$ be the probability of the process arriving at mode I, from either mode 1, 2, or 3, at the end of cycle T. Here, I may represent the modes R or F, and T may take on the values

j, j+1, j+2, ...

The probability of the process arriving at mode R at the end of cycle T, from either mode 1, 2 or 3, is given by

$$\begin{aligned} P_R{}^*(T) &= \text{Prob. (mode 1 in cycle T-1} \rightarrow \text{mode R in cycle T)} \\ &+ \text{Prob. (mode 2 in cycle T-1} \rightarrow \text{mode R in cycle T)} \\ &+ \text{Prob. (mode 3 in cycle T-1} \rightarrow \text{mode R in cycle T)} \end{aligned}$$

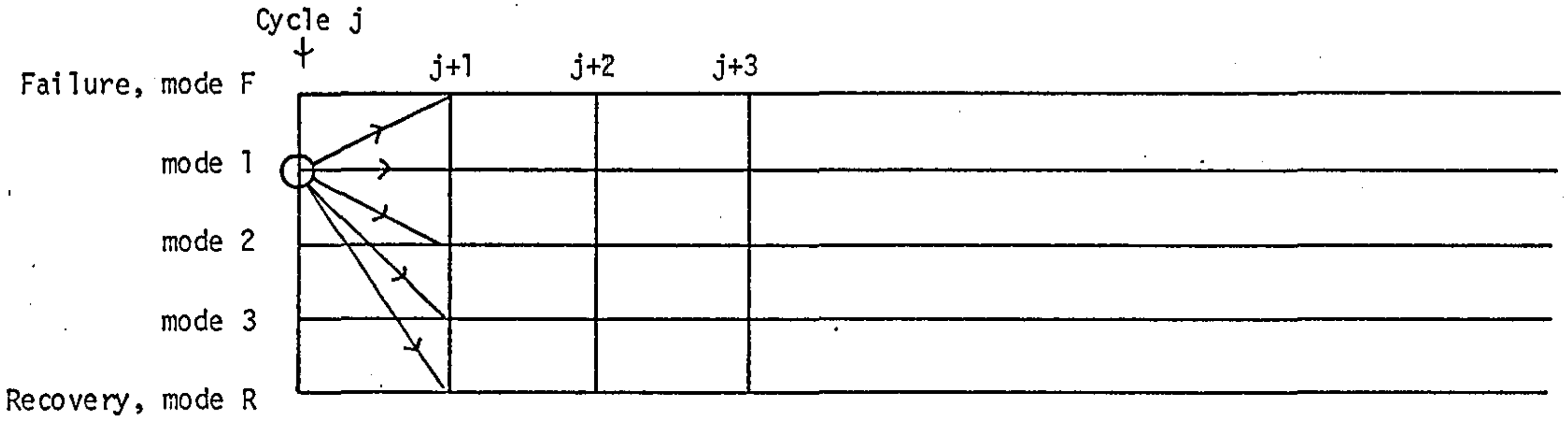


FIGURE 5.02

Transitions from one mode to another

Therefore

$$P_R^*(T) = P_1(T-1)\ p_{1R} + P_2(T-1)\ p_{2R} + P_3(T-1)\ p_{3R} \qquad (5.24)$$

Similarly the probability of the process arriving at mode F at the end of cycle T, from either mode 1, 2 or 3, is given by

$$P_F^*(T) = P_1(T-1)\ p_{1F} + P_2(T-1)\ p_{2F} + P_3(T-1)\ p_{3F} \qquad (5.25)$$

The probability of the process being in one of the modes: 1, 2 or 3, at the end of cycle T, is related to the probabilities at the end of cycle T-1, by the equations:

$$P_1(T) = P_1(T-1)\ p_{11} + P_2(T-1)\ p_{21} + P_3(T-1)\ p_{31}$$

$$P_2(T) = P_1(T-1)\ p_{12} + P_2(T-1)\ p_{22} + P_3(T-1)\ p_{32}$$

$$P_3(T) = P_1(T-1)\ p_{13} + P_2(T-1)\ p_{23} + P_3(T-1)\ p_{33}$$

It should be noted that the purpose of this analysis, is to determine the probability of the process eventually entering the failure mode, given that it has entered either mode 1, 2 or 3 at some stage. Hence the probabilities of the process going from mode R to 1, 2 or 3, or of going from mode F to 1, 2 or 3, are not required here.

The above three equations may be written in matrix form to give

$$\begin{bmatrix} P_1(T) \\ P_2(T) \\ P_3(T) \end{bmatrix} = \begin{bmatrix} p_{11} & p_{21} & p_{31} \\ p_{12} & p_{22} & p_{32} \\ p_{13} & p_{23} & p_{33} \end{bmatrix} \begin{bmatrix} P_1(T-1) \\ P_2(T-1) \\ P_3(T-1) \end{bmatrix}$$

for T = j+1, j+2, ...

Therefore

$$\underline{P}(T) = A\,\underline{P}(T-1)$$

where

$$\underline{P}(T) = \begin{bmatrix} P_1(T) \\ P_2(T) \\ P_3(T) \end{bmatrix} \qquad \text{and} \qquad A = \begin{bmatrix} p_{11} & p_{21} & p_{31} \\ p_{12} & p_{22} & p_{32} \\ p_{13} & p_{23} & p_{33} \end{bmatrix}$$

Hence

$$\begin{aligned} \underline{P}(T) &= A^2\,\underline{P}(T-2) \\ &= A^3\,\underline{P}(T-3) \\ &\;\;\vdots \\ &= A^i\,\underline{P}(T-i) \end{aligned}$$

for i = T-j, T-j+1, T-j+2, ...

Therefore, putting i=T-j gives

$$\underline{P}(T) = A^{T-j}\,\underline{P}(j) \tag{5.26}$$

or

$$\underline{P}(T-1) = A^{T-j-1}\,\underline{P}(j)$$

for T = j+1, j+2, j+3, ...

$$\therefore \sum_{T=j+1}^{\infty} \underline{P}(T-1) = \sum_{T=j+1}^{\infty} A^{T-1-j} \underline{P}(j)$$

$$= \left(\sum_{T=j+1}^{\infty} A^{T-1-j}\right) \underline{P}(j)$$

$$\sum_{T=j+1}^{\infty} \underline{P}(T-1) = \left(\sum_{T=0}^{\infty} A^{T}\right) \underline{P}(j) \qquad (5.27)$$

(It is assumed here that the infinite series is convergent).

Now let

$$S_n = \sum_{T=0}^{n} A^T$$

so that

$$S_n = I + A + A^2 + \ldots\ldots + A^n.$$

Then

$$A\, S_n = A + A^2 + \ldots\ldots + A^{n+1}$$

Subtracting the second expression from the first gives

$$(I-A)\, S_n = I - A^{n+1}$$

Now assume that I-A is invertible. Then

$$S_n = (I-A)^{-1} (I-A^{n+1})$$

$$= (I-A)^{-1} - (I-A)^{-1} A^{n+1}.$$

From the definition of s_n,

$$\sum_{T=0}^{\infty} A^T = \lim_{n\to\infty} s_n$$

$\therefore$

$$\sum_{T=0}^{\infty} A^T = \lim_{n\to\infty} \{(I-A)^{-1} - (I-A^{-1})\ A^{n+1}\}$$

$$\sum_{T=0}^{\infty} A^T = (I-A)^{-1} \qquad (5.28)$$

if it is assumed that $A^n \to$ the zero matrix as $n\to\infty$. (This assumption will be justified at a later stage in this section). If (I-A) is not invertible, the infinite series must be summed in some other way.

From equations 5.27 and 5.28

$$\sum_{T=j+1}^{\infty} \underline{P}(T-1) = (I-A)^{-1}\ \underline{P}(j) \qquad (5.29)$$

Now let P_F be the probability of the process eventually reaching the failure mode (mode F), given that it has entered either mode 1, 2 or 3, during some cycle j. Then

$$P_F = \text{Prob.} \left[\begin{array}{l} \text{The process reaches mode F from either mode 1, 2 or 3,} \\ \text{at the end of cycle j+1} \\ \text{or} \\ \text{The process reaches mode F from either mode 1, 2 or 3,} \\ \text{at the end of cycle j+2} \\ \text{or} \\ \cdot \\ \cdot \\ \cdot \end{array} \right]$$

It has been shown that the process is likely to remain in mode F for a large number of cycles, once it has entered this mode. (See Section 5.04). Now consider the event in which the process reaches mode F from either mode 1, 2 or 3, at the end of some cycle T. Clearly the process will settle in mode F, once it arrives there, so these events are disjoint for different values of T, i.e. if the process enters mode F from either mode 1, 2 or 3 at the end of cycle T, this cannot be repeated at the end of cycle T+1. Hence the above equation for P_F becomes

$$P_F = \sum_{T=j+1}^{\infty} \text{Prob. (The process reaches mode F from either mode 1, 2 or 3, at the end of cycle T).}$$

$$P_F = \sum_{T=j+1}^{\infty} P_F^*(T) \qquad (5.30)$$

(from the definition of $P_F^*(T)$). But, from equation 5.25

$$P_F^*(T) = P_1(T-1)\, p_{1F} + P_2(T-1)\, p_{2F} + P_3(T-1)\, p_{3F}.$$

$$\therefore$$

$$P_F = \sum_{T=j+1}^{\infty} (P_1(T-1)\, p_{1F} + P_2(T-1)\, p_{2F} + P_3(T-1)\, p_{3F})$$

$$= p_{1F} \sum_{T=j+1}^{\infty} P_1(T-1) + p_{2F} \sum_{T=j+1}^{\infty} P_2(T-1) + p_{3F} \sum_{T=j+1}^{\infty} P_3(T-1)$$

$$= (p_{1F}, p_{2F}, p_{3F}) \sum_{T=j+1}^{\infty} \begin{bmatrix} P_1(T-1) \\ P_2(T-1) \\ P_3(T-1) \end{bmatrix}$$

$$= (p_{1F}, p_{2F}, p_{3F}) \sum_{T=j+1}^{\infty} \underline{P}\ (T-1) \qquad (5.31)$$

But from equation 5.29

$$\sum_{T=j+1}^{\infty} \underline{P}\ (T-1) = (I-A)^{-1}\ \underline{P}\ (j)$$

$$\therefore\ P_F = (p_{1F}, p_{2F}, p_{3F})\ (I-A)^{-1}\ \underline{P}(j) \qquad (5.32)$$

Convergence of $\sum_{T=0}^{\infty} A^T$

Clearly P_F is the probability of a certain event occurring, so

$$0 \leq P_F \leq 1.$$

Hence, assuming that p_{1F}, p_{2F} and p_{3F} are non zero, it is clear from equation 5.31 that

$$\sum_{T=j}^{\infty} \underline{P}\ (T-1)$$

must be a vector with finite components (as p_{1F}, p_{2F}, p_{3F}, $P_1(T)$, $P_2(T)$, $P_3(T)$ are all probabilities and are ≥ 0). Hence from equation 5.27

$$(\sum_{T=0}^{\infty} A^T)\ \underline{P}(j)$$

must be a vector with finite components. This is true whether the process is in mode 1, 2 or 3.at the end of cycle j. But

$$\underline{P}(j) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \text{ or } \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

depending on whether the appropriate mode is 1, 2 or 3, respectively. Therefore the three vectors

$$(\sum_{T=0}^{\infty} A^T)\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad (\sum_{T=0}^{\infty} A^T)\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \text{and} \quad (\sum_{T=0}^{\infty} A^T)\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

must each be finite. The matrix whose columns are formed from these three vectors must also be finite, hence

$$(\sum_{T=0}^{\infty} A^T)\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

is a finite matrix, and the components of

$$\sum_{T=0}^{\infty} A^T$$

must each be finite. Therefore

$$\sum_{T=0}^{\infty} A^T$$

is a convergent series and $A^T \to 0$ as $T \to \infty$, as assumed above.

An expression for the probability P_F of the process eventually reaching the failure mode is given above, in equation 5.32. A similar expression will now be derived for the probability P_R, of the process eventually reaching the recovery mode.

P_R may be expressed in terms of the probabilities $P_R^*(T)$, of the process entering mode R from either mode 1, 2 or 3, by the equation:

$$P_R = \sum_{T=j+1}^{\infty} P_R^* \ (T)$$

(compare with equation 5.30). But, from equation 5.24,

$$P_R^*(T) = P_1(T-1)\ p_{1R} + P_2(T-1)\ p_{2R} + P_3(T-1)\ p_{3R}$$

$$\therefore \quad P_R = p_{1R} \sum_{T=j+1}^{\infty} P_1(T-1) + p_{2R} \sum_{T=j+1}^{\infty} P_2(T-1) + p_{3R} \sum_{T=j+1}^{\infty} P_3(T-1)$$

$$= (p_{1R},\ p_{2R},\ p_{3R}) \sum_{T=j+1}^{\infty} \begin{bmatrix} P_1(T-1) \\ P_2(T-1) \\ P_3(T-1) \end{bmatrix}$$

$$= (p_{1R},\ p_{2R},\ p_{3R}) \sum_{T=j+1}^{\infty} \underline{P}\,(T-1)$$

But, from equation 5.29

$$\sum_{T=j+1}^{\infty} \underline{P}(T-1) = (I-A)^{-1} \underline{P}(j)$$

(assuming that I-A is invertible). Hence

$$P_R = (p_{1R}, p_{2R}, p_{3R})\,(I-A)^{-1}\,\underline{P}(j) \qquad (5.33)$$

Now suppose that it is given that the process was in mode I at the end of cycle j. Then, from the definition of $\underline{P}(T)$, it is clear that

$$\underline{P}(j) = \begin{bmatrix} f_1(I) \\ f_2(I) \\ f_3(I) \end{bmatrix}$$

where

$$f_J(I) = \begin{cases} 1 & \text{if } I=J \\ 0 & \text{otherwise} \end{cases}$$

Let $P_{R/I}$ be the probability of the process eventually reaching the recovery mode, given that it was in mode I at the end of cycle j.

Then, from equation 5.33,

$$P_{R/I} = (p_{1R}, p_{2R}, p_{3R})(I-A)^{-1} \begin{bmatrix} f_1(I) \\ f_2(I) \\ f_3(I) \end{bmatrix}$$

Also, from equation 5.32

$$P_{F/I} = (p_{1F}, p_{2F}, p_{3F})(I-A)^{-1}\begin{bmatrix} f_1(I) \\ f_2(I) \\ f_3(I) \end{bmatrix}$$

where $P_{F/I}$ is the probability of the process eventually reaching mode F, given that it was in mode I at the end of cycle j. The above two equations may be combined to give the equation

$$P_{K/I} = (p_{1K}, p_{2K}, p_{3K})(I-A)^{-1}\begin{bmatrix} f_1(I) \\ f_2(I) \\ f_3(I) \end{bmatrix} \qquad (5.34)$$

where K may represent the modes R and F, and I may take the values 1, 2 or 3.

5.06 Evaluation of the Transition Probabilities p_{IJ}

The transitions of a System 1 detection process, from one mode to another, are dictated by several factors. These transitions depend on the costs associated with the stored vectors, the data sequence transmitted and the sequence of noise samples. There appears to be no feasible analytical method for evaluating the transition probabilities, but they may be found approximately using computer simulations.

During the operation of System 1, the appearance of a pair of vectors which are merged with a small separation ($\leq 10^{-3}$say), is a fairly rare event. This means that a very large amount of simulation testing would be required, to obtain a reasonable number of cases in which such pairs of vectors were formed. The simulation tests were therefore initialized with an artificially contrived set of vectors and costs, containing a pair of vectors which were merged with separation zero. The following method appears to be a reasonable one, for constructing such a set of vectors and costs.

The detection process is started up in the usual way, as described in Section 3.08. Then after a few data elements (say 15 or 20) have been detected, the values for one of the stored vectors and its cost , are changed. These values are replaced by the corresponding values for one of the other vectors, so that two identical vectors and costs are present in the system. The simulation test is then completed, upon noting which mode the process has entered at the end of the following cycle.

When a number of such simulation tests have been completed, the proportion of tests in which the process has moved from one particular mode to another, gives an estimate of the corresponding transition probability. The accuracy of this estimate naturally increases with the number of simulation tests performed.

Let the four vectors stored at the end of the j+1st. cycle of the process, be denoted $\underline{Q}_j(I)$, with costs $u_j(I)$, for I = 1, 2, 3 and 4. Now suppose that it is desired to set the detection process in mode I, at the end of cycle j. (Mode I being that in which the vectors $\underline{Q}_j(I)$ and $\underline{Q}_j(I+1)$ are merged with zero separation). The method described above allows two alternatives for setting the process in this mode. Either the values for $\underline{Q}_j(I)$ and $u_j(I)$ can be replaced by those for $\underline{Q}_j(I+1)$ and $u_j(I+1)$, or the values for $\underline{Q}_j(I+1)$

and $u_j(I+1)$ can be replaced by those for $\underline{Q}_j(I)$ and $u_j(I)$. Let the modes of the system for these two cases, be denoted mode IA and mode IB.

For the estimation of the transition probability p_{IJ}, an equal number of simulation tests were carried out, with the process initialized in mode IA and mode IB. These tests provided estimates of the transition probabilities $p_{IA,J}$ and $p_{IB,J}$, of the process moving from mode IA to J and mode IB to J, respectively. The estimate for p_{IJ} was then taken to be the average of the estimates for $p_{IA,J}$ and $p_{IB,J}$.

The simulation tests described in this section were performed on System 1 with a binary data signal. Channel E, 4 stored vectors (at the start of each cycle) and a value of 0.178 for the noise standard deviation, were used throughout. Note that 0.178 is the value of the noise standard deviation, which would cause an error rate of 0.004, if the detection process was working normally without any merged vectors. The number N, of components of the vectors, at the start of each cycle, was fixed at eleven.

Table 5.01 gives the estimates obtained for the transition probabilities $p_{IA,J}$, $p_{IB,J}$ and p_{IJ} for a case where 100 trials were performed for each of the six starting modes (i.e. the six modes: 1A, 1B, 2A, 2B, 3A and 3B)

Note that the four vectors stored at the start of the j+1 st. cycle of the process, are denoted

$\underline{Q}_j(1)$, $\underline{Q}_j(2)$, $\underline{Q}_j(3)$ and $\underline{Q}_j(4)$

I	$p_{IA,F}$	$p_{IB,F}$	p_{IF}
1	0.46	0.80	0.630
2	0.04	0.18	0.110
3	0.00	0.01	0.005

I	$p_{IA,1}$	$p_{IB,1}$	p_{I1}
1	0.16	0.08	0.120
2	0.06	0.02	0.040
3	0.00	0.03	0.015

I	$p_{IA,2}$	$p_{IB,2}$	p_{I2}
1	0.14	0.05	0.095
2	0.06	0.08	0.070
3	0.00	0.03	0.015

I	$p_{IA,3}$	$p_{IB,3}$	p_{I3}
1	0.10	0.03	0.065
2	0.63	0.51	0.570
3	0.24	0.32	0.280

I	$p_{IA,R}$	$p_{IB,R}$	p_{IR}
1	0.14	0.04	0.090
2	0.21	0.21	0.210
3	0.76	0.61	0.685

Tests performed with channel E, k=4, σ = 0.178 and N=11.

TABLE 5.01

Results of simulation tests for evaluating the transition probabilities p_{IJ}, with 100 trials in ascending order of costs. Hence, from the definitions of modes IA and IB, given above, it can be seen that the merged pair of vectors will have a higher cost in mode IA than in mode IB. Whichever the initial mode in the simulation tests, the merged pair of vectors will be extended to two merged pairs in the following cycle of the process (see Section 5.03). If a simulation test is started from mode IA, it should usually be the case that the costs for the two merged pairs of vectors, are greater than those occurring from an initial mode IB. The decision rule is therefore more likely to select both pairs of merged vectors, for the case where the initial mode is IB. (The decision rule selects the k vectors with lowest costs). The probability of the process moving to the failure mode, should therefore be greater from the initial mode IB, than from IA. This is supported by the results shown in Table 5.01.

The results given in Table 5.01, are for a case where 100 trials were carried out for each of the initial modes 1A, 1B, 2A, 2B, 3A and 3B. The result of each trial is, of course, the mode that the process has moved into after one cycle. Tables 5.02, 5.03 and 5.04 give the results of similar simulation tests, where the number n of trials for each starting mode, was 500, 1901 and 20001 respectively. Then, from the results of each of the tables 5.01-5.04, the probabilities of the process eventually reaching the recovery and failure modes, were calculated as in Section 5.05.

The estimated probabilities $P_{K/I}$, of the process eventually reaching mode K given that it had entered mode I, are given in Table 5.05. These estimates are given for the four different values of n, corresponding to tables 5.01-5.04. (n is, of course, the number

I	$p_{IA,F}$	$p_{IB,F}$	p_{IF}
1	0.408	0.826	0.617
2	0.034	0.160	0.097
3	0.010	0.010	0.010

I	$p_{IA,1}$	$p_{IB,1}$	p_{I1}
1	0.194	0.068	0.131
2	0.038	0.028	0.033
3	0.000	0.008	0.004

I	$p_{IA,2}$	$p_{IB,2}$	p_{I2}
1	0.130	0.032	0.081
2	0.052	0.082	0.067
3	0.002	0.038	0.020

I	$p_{IA,3}$	$p_{IB,3}$	p_{I3}
1	0.116	0.036	0.076
2	0.618	0.506	0.562
3	0.226	0.338	0.282

I	$p_{IA,R}$	$p_{IB,R}$	p_{IR}
1	0.152	0.038	0.095
2	0.258	0.224	0.241
3	0.762	0.606	0.684

Tests performed with channel E, k=4, $\sigma = 0.178$ and N=11.

TABLE 5.02

Results of simulation tests for evaluating the transition probabilities p_{IJ}, with 500 trials.

I	$p_{IA,F}$	$p_{IB,F}$	p_{IF}
1	0.396	0.833	0.615
2	0.028	0.155	0.092
3	0.006	0.008	0.007

I	$p_{IA,1}$	$p_{IB,1}$	p_{I1}
1	0.216	0.080	0.148
2	0.032	0.032	0.032
3	0.002	0.005	0.004

I	$p_{IA,2}$	$p_{IB,2}$	p_{I2}
1	0.122	0.033	0.077
2	0.055	0.078	0.067
3	0.011	0.027	0.019

I	$p_{IA,3}$	$p_{IB,3}$	p_{I3}
1	0.129	0.025	0.077
2	0.620	0.519	0.570
3	0.229	0.335	0.282

I	$p_{IA,R}$	$p_{IB,R}$	p_{IR}
1	0.137	0.028	0.083
2	0.264	0.216	0.240
3	0.752	0.625	0.689

Tests performed with channel E, k=4, σ = 0.178 and N=11.

TABLE 5.03

Results of simulation tests for evaluating the transition probabilities p_{IJ}, with 1901 trials.

I	$p_{IA,F}$	$p_{IB,F}$	p_{IF}
1	0.404	0.836	0.620
2	0.028	0.155	0.092
3	0.004	0.007	0.005

I	$p_{IA,1}$	$p_{IB,1}$	p_{I1}
1	0.213	0.087	0.150
2	0.028	0.033	0.031
3	0.002	0.003	0.003

I	$p_{IA,2}$	$p_{IB,2}$	p_{I2}
1	0.129	0.024	0.076
2	0.063	0.086	0.074
3	0.011	0.031	0.021

I	$p_{IA,3}$	$p_{IB,3}$	p_{I3}
1	0.130	0.026	0.078
2	0.617	0.521	0.569
3	0.239	0.331	0.285

I	$p_{IA,R}$	$p_{IB,R}$	p_{IR}
1	0.125	0.027	0.076
2	0.264	0.205	0.234
3	0.744	0.628	0.686

Tests performed with channel E, k=4, σ = 0.178 and N=11.

TABLE 5.04

Results of simulation tests for evaluating the transition probabilities p_{IJ}, with 20001 trials.

n	$P_{R/1}$	$P_{R/2}$	$P_{R/3}$
100	0.264	0.834	0.974
500	0.275	0.857	0.978
1901	0.265	0.867	0.984
20001	0.258	0.868	0.986

n	$P_{F/1}$	$P_{F/2}$	$P_{F/3}$
100	0.736	0.166	0.026
500	0.725	0.143	0.022
1901	0.735	0.134	0.017
20001	0.743	0.132	0.014

TABLE 5.05

The calculated probabilities $P_{I/K}$ for different numbers n, of trials.

of trials carried out with each of the six starting modes, to obtain the estimates of the transition probabilities p_{IJ}).

The estimates of the probabilities $P_{I/K}$ are, of course, subject to statistical fluctuation. It is not a straightforward matter to obtain confidence limits for these estimates, but the accuracy must improve as the number n of trials is increased. If n is large enough, so that increasing its value does not significantly change the estimates of $P_{I/K}$, it seems reasonable that enough trials have been carried out to give fairly accurate results. The estimates of any particular probability $P_{K/I}$, for n = 1901 and 20001, differ by no more than 0.01, in Table 5.05. It might be assumed, therefore, that the estimates corresponding to n = 20001, are accurate within a tolerance of ±0.01.

Now consider again the results given in Table 5.05. It may be seen from these results, that the probabilities of the process reaching the modes R and F, sum to unity, irrespective of whether the process was initially set in mode 1, 2 or 3. It appears, therefore, that the system can not remain within the modes 1, 2 and 3 indefinitely. This is confirmed by the following theoretical analysis.

From equation 5.26,

$$\underline{P}(T) = A^{T-j}\,\underline{P}(j)$$

or

$$\begin{bmatrix} P_1(T) \\ P_2(T) \\ P_3(T) \end{bmatrix} = A^{T-j} \begin{bmatrix} P_1(T-1) \\ P_2(T-1) \\ P_3(T-1) \end{bmatrix}$$

where $P_I(T)$ is the probability that the process will be in mode I, at the end of cycle T, for I = 1, 2, and 3. It was shown in Section 5.05, that A^T tends to the zero matrix as $T \to \infty$. Hence $P_I(T) \to 0$ as $T \to \infty$, for I = 1, 2 and 3, i.e. the probability of the process being in any of the modes: 1, 2 and 3 at the end of cycle T, tends to zero as T becomes large. The process must therefore leave the modes 1, 2 and 3, and eventually settle in mode R or mode F.

For the matrices A, corresponding to the transition probabilities given in tables 5.01-5.04, it was found that the elements of A^T were effectively zero, for $T \geq 20$. In fact the two sums:

$$\sum_{T=0}^{\infty} A^T \qquad \text{and} \qquad \sum_{T=0}^{20} A^T$$

were found to be the same, within a tolerance of 10^{-6} for the elements of the matrices. Hence, from equation 5.26, it can be seen that $\underline{P}(T)$ has all of its components effectively equal to zero, for $T \geq j + 20$. i.e. the probability of the process leaving modes 1, 2 and 3, within twenty cycles of entering one of these modes, is effectively equal to unity. The process is therefore bound to settle in either mode R or mode F, within about twenty cycles of a merged pair of vectors appearing. (Note that this result applies to a particular situation in which the number k of stored vectors was equal to four).

5.07 The Direct Evaluation of $P_{K/I}$ by Computer Simulation

In Section 5.05, a method was given for evaluating the probability $P_{K/I}$, that the System 1 detection process will eventually enter mode K, given that it has entered mode I. I may take on the values 1, 2 and 3, and K may represent either mode R or mode F. This method calculates the probabilities $P_{K/I}$, from the probabilities p_{IJ}, of the process moving from mode I to mode J in one cycle. The problem with this approach is that of obtaining accurate estimates of the transition probabilities p_{IJ}.

The elements of the matrix A, in equation 5.34, are from the set of estimates of the transition probabilities p_{IJ}. It is possible that errors in the elements of A, may cause larger errors in the elements of $(I-A)^{-1}$. It can then be seen from equation 5.34, that large errors may be present in the estimated values of the probabilities $P_{K/I}$.

Another problem with the method of Section 5.05, is that it is very complicated for values of k which are much greater than four. (k is the number of vectors stored at the start of each cycle). For larger values of k, the number of modes of the trellis, of the type shown in Figure 5.03, becomes alarmingly large. This method for calculating $P_{K/I}$, is however useful in that the analysis involved, gives some insight into the behaviour of the detection process.

The probabilities $P_{K/I}$ have also been estimated directly from simulation results. As with the tests described in Section 5.06, these latter tests were carried out with the process placed in each of the modes 1A, 1B, 2A, 2B, 3A and 3B. The detection process was then continued for as many cycles as were needed, for it to enter either mode R or mode F. (i.e. the recovery of the failure mode).

Consider a number n of simulation trials in which the process is placed in mode 1A. The proportions of trials in which the modes R and F eventually occurred, then give estimates of the probabilities $P_{R/1A}$ and $P_{F/1A}$. (i.e. the probabilities of the process eventually reaching the modes R and F, given that it was placed in mode 1A at some stage). The same number n of simulation trials, were carried out to provide estimates of $P_{R/1B}$ and $P_{F/1B}$, and the estimates for $P_{R/1}$ and $P_{F/1}$ were defined by

$$P_{R/1} = \frac{P_{R/1A} + P_{R/1B}}{2}$$

and

$$P_{F/1} = \frac{P_{F/1A} + P_{F/1B}}{2}$$

This procedure was repeated for the probabilities: $P_{R/2}$, $P_{R/3}$, $P_{F/2}$ and $P_{F/3}$.

The results of the simulation tests for estimating the probabilities $P_{K/I}$, are given in Tables 5.06-5.08. These results cover situations with different values of k, different noise levels and two different channels. (k is, of course, the number of vectors stored at the start of each cycle of the detection process). For cases where k is greater than four, the modes of operation of the process, are defined in basically the same way as before. Mode R is the mode in which there are no merged vectors, and F is the mode in which the vectors form into merged pairs with zero separation. Mode I is defined to be that in which only the vectors $\underline{Q}_j(I)$ and $\underline{Q}_j(I+1)$, are merged with zero separation, for I = 1, 2, 3, ... (Note that the vector with K th. largest cost is denoted $\underline{Q}_j(K)$, for K = 1, 2, ..., k).

	I	$P_{F/IA}$	$P_{F/IB}$	$P_{F/I}$
k=4	1	0.4692	0.9269	0.6981
	2	0.0731	0.1577	0.1154
	3	0.0038	0.0269	0.0154
k=6	1	0.3692	0.8731	0.6212
	2	0.0654	0.2308	0.1481
	3	0.0038	0.0154	0.0096
k=8	1	0.2769	0.8038	0.5404
	2	0.0539	0.1538	0.1039
	3	0.0038	0.0077	0.0058
k=16	1	0.2846	0.8231	0.5539
	2	0.0423	0.2308	0.1366
	3	0.0	0.0115	0.0058

TABLE 5.06

Variation of $P_{F/I}$ with the value k, for System 1, channel E and $\sigma = 0.178$. 260 trials.

	I	$P_{F/IA}$	$P_{F/IB}$	$P_{F/I}$
	1	0.342	0.931	0.637
σ=0.136	2	0.023	0.104	0.064
	3	0.000	0.000	0.000
	1	0.396	0.873	0.635
σ=0.178	2	0.065	0.231	0.148
	3	0.004	0.015	0.010
	1	0.454	0.665	0.560
σ=0.282	2	0.169	0.296	0.233
	3	0.050	0.100	0.075

TABLE 5.07

Variation of $P_{F/I}$ with the value of σ, for System 1, channel E, and k=6. 260 trials.

	I	$P_{F/IA}$	$P_{F/IB}$	$P_{F/I}$
k=4	1	0.742	0.919	0.831
	2	0.081	0.239	0.160
	3	0.000	0.004	0.002
k=6	1	0.481	0.835	0.658
	2	0.073	0.289	0.181
	3	0.004	0.015	0.010
k=8	1	0.458	0.781	0.619
	2	0.039	0.269	0.154
	3	0.008	0.015	0.012
k=16	1	0.373	0.765	0.569
	2	0.042	0.323	0.183
	3	0.000	0.012	0.006

TABLE 5.08

Variation of $P_{F/I}$ with the value of k, for System 1, channel J and $\sigma = 0.0762$. 260 trials.

It was shown in Section 5.06, that the System 1 detection process with k=4 is very likely to enter either the recovery or failure mode, given that it has entered mode 1, 2 or 3 at some stage. The same analysis can be applied to any case where k is even, therefore

$$P_{R/I} + P_{F/I} = 1$$

for I = 1, 2 and 3. i.e. the probabilities of the process going from mode I to the recovery mode, and the failure mode, should sum to unity. (This is supported by simulation results). Hence only the probabilities of the process entering mode F, are given in tables 5.06-5.09.

Table 5.06 shows how the probabilities $P_{F/I}$, vary with the value of k, for System 1, channel E and a value of 0.178 for the noise standard deviation σ. (This is the value of σ which will give an error rate of 0.004 under the given conditions, when there are no merged vectors present in the system). The number N of components of the stored vectors was fixed at eleven, as was the case for all simulation tests described in this section. Each probability $P_{F/I}$ was calculated as explained above, with 260 simulation trials carried out for the estimation of each of the probabilities $P_{F/IA}$ and $P_{F/IB}$. (i.e. n = 260).

It can be seen from Table 5.06, that the probabilities: $P_{F/1}$, $P_{F/2}$ and $P_{F/3}$, do not change dramatically with the value of k, for the situation tested. These probabilities do, however, decrease slightly as k is increased from 4 to 16. It is also evident from this table, that the probabilities $P_{F/I}$, decrease rapidly as I increases from 1 to 3, for any of the values of k.

Table 5.07 demonstrates the way in which the probabilities: $P_{F/1}$, $P_{F/2}$ and $P_{F/3}$, vary with the noise level in the system. The corresponding simulation tests were carried out on System 1, with channel E and a value of 6, for k. As before, N was fixed at eleven, and 260 simulation trials were carried out, for the estimate of each of the probabilities $P_{F/IA}$ and $P_{F/IB}$. The three values of the noise standard deviation σ, used in these tests were 0.136, 0.178 and 0.282. These σ values are ones which would cause error rates of 10^{-4}, 0.004 and 10^{-1} respectively under the given conditions, if there were no merged vectors present in the system.

It can be seen from Table 5.07, that the probability $P_{F/1}$ does not vary greatly with the value of σ. The probabilities $P_{F/2}$ and $P_{F/3}$ do, however, increase quite noticeably as σ increases.

Table 5.08 shows the results of simulation tests on System 1, with channel J and σ = 0.0762. This is the value of σ which would cause an error rate of 0.004, under the given conditions, if there were no merged vectors present in the system. These simulation tests were the same as those corresponding to Table 5.06, except where stated. From Tables 5.06 and 5.08, it can be seen that variations in the value of k have a similar affect on the probabilities $P_{F/I}$, for both channel E and channel J.

It has been observed from simulation results that, if a vector $\underline{Q}_j(I)$ has a large cost in relation to the other stored vectors, then the vectors stemming from $\underline{Q}_j(I)$ will have large costs. The decision rule for System 1 is such, that the k vectors with smallest cost are selected from a set of mk vectors, during each cycle of the process. Hence it can be seen that vectors stemming from such a vector $\underline{Q}_j(I)$, are likely to be deleted from the system.

Now consider System 1 when operating with an even number k, of vectors stored at the start of each cycle. Suppose that the process enters mode I in the j+1 st. cycle, so that the vectors $\underline{Q}_j(I)$ and $\underline{Q}_j(I+1)$ are merged with some small separation. i.e. the vectors with I th. smallest cost and I+1 st. smallest cost , are merged in cycle j+1. (I may have any value in the range 1, 2, ..., k-1). If I is fairly large, the costs for these two merged vectors will be among the greatest costs present in the j+1 st. cycle of the process. Then, from the above discussion, it can be seen that the vectors stemming from these merged vectors, are likely to be deleted, thus allowing the process to enter the recovery mode. Hence it should be expected that the probability $P_{F/I}$, of the process eventually reaching the failure mode given that it has entered mode I, must decrease as I increases. This is substantiated for I = 1, 2 and 3, by the results given in Tables 5.06-5.08.

It can be seen from Tables 5.06-5.08, that the decrease in $P_{F/I}$ as I increases, is fairly rapid apart from the case with channel E and $\sigma = 0.282$. In all of the other cases tested, the value of σ was chosen such that it would give an error rate of 0.004 or less, if there were no merged vectors present in the system. These results suggest that the probabilities $P_{F/I}$ are very small for $I > 3$, when the detection process is working at error rates of 0.004 or less. (Note that $P_{F/I}$ is only defined for $I \leq 3$ if k=4).

In Sections 5.05 and 5.07, two methods were given for evaluating the probabilities: $P_{F/1}$, $P_{F/2}$ and $P_{F/3}$. Table 5.05 gives the results of the former method, for channel E with k=4 and $\sigma = 0.178$. Results for the latter method may be found in Table 5.06, but the confidence limits for these results are not really close enough to

allow a comparison of the two methods. (See Section 5.08 for a derivation of the confidence limits, for the method described in Section 5.07). The simulation tests corresponding to Table 5.06, were therefore repeated with $n=10^4$. i.e. 10^4 trials were carried out to provide estimates of each of the probabilities $P_{F/IA}$ and $P_{F/IB}$. The results of these tests are given in Table 5.09.
A comparison of Tables 5.05 and 5.09, shows that the two methods of calculating the probabilities $P_{F/I}$, are in reasonable agreement. (The results for $n = 20001$ should be used, from Table 5.05, as these have the closest confidence limits).

5.08 Confidence Limits for the Estimates of $P_{K/I}$

Consider a series of n independent trials, each of which has only two outcomes: success and failure. Let p be the probability of failure, so that the probability of success is 1-p.

Let p* be the proportion of failures in the n trials. Then, if np and n(1-p) are reasonably large (greater than 5 say), it may be shown that the distribution of the random variable p*, is approximately $N\ (p, \frac{p(1-p)}{n})$.

i.e. the distribution is approximately normal, with mean p and variance = p(1-p)/n. [50]

It may also be shown that, for any normally distributed random variable with mean μ and variance σ^2, there is a 95% probability that

$$x - 1.96\sigma \leq \mu \leq x + 1.96\sigma$$

for any sample x, taken from the distribution [50]. Applying this

I	$P_{F/IA}$	$P_{F/IB}$	$P_{F/IC}$
1	0.511	0.905	0.708
2	0.058	0.200	0.129
3	0.002	0.009	0.006

TABLE 5.09

Values of the probabilities $P_{F/I}$ for System 1, channel E, k=4 and σ = 0.178. 10^4 trials.

result to the situation described above, gives

$$\text{Prob.}(p^* - 1.96\sqrt{\frac{p(1-p)}{n}} \leq p \leq p^* + 1.96\sqrt{\frac{p(1-p)}{n}}) \simeq 0.95$$

If the number n of trials is fairly large, p will be quite close to p* and

$$p(1-p) \simeq p^* (1-p^*)$$

$$\therefore\ \text{Prob.}(p^* - 1.96\sqrt{\frac{p^*(1-p^*)}{n}} \leq p \leq p^* + 1.96\sqrt{\frac{p^*(1-p^*)}{n}}) \simeq 0.95$$

(5.35)

Hence, if p* is used as an estimate of p, the 95% confidence limits for the estimate are approximately

$$\pm 1.96\sqrt{\frac{p^*(1-p^*)}{n}}$$

For the simulation tests corresponding to Tables 5.06-5.08, the number n of trials carried out for the estimation of each of the probabilities $P_{F/IA}$ and $P_{F/IB}$, was 260. Each trial was independent of the others, and had two possible outcomes. The process could either enter the recovery mode, or the failure mode. Let p* denote the estimate of the probability of the process entering the failure mode, in some such group of 260 trials. Then, from equation 5.35, the 95% confidence limits for p* are

$$\pm 1.96\sqrt{\frac{p^*(1-p^*)}{260}}$$

p* can, however, only be assumed to be normally distributed if np* and n(1-p*) are reasonably large (≥ 5 say). Hence, taking n = 260, the above analysis is valid if p* is ≥0.02.

It may readily be shown by differentiation, that p*(1-p*) takes its maximum value when p* = ½. Therefore

$$1.96\sqrt{\frac{p^*(1-p^*)}{260}} \leq 1.96\sqrt{\frac{\frac{1}{2} \times \frac{1}{2}}{260}}$$

$$\leq 0.061$$

Hence the 95% confidence limits for the values of $P_{F/IA}$ and $P_{F/IB}$, in Tables 5.06-5.08, are no more than ±0.061. (This only applies when the probabilities are ≥ 0.02).

For the simulation tests corresponding to Table 5.09, $n=10^4$. Hence, from equation 5.35, the 95% confidence limits for these values of $P_{F/IA}$ and $P_{F/IB}$, are

$$\pm 1.96\sqrt{\frac{p^*(1-p^*)}{10^4}}$$

But $$1.96\sqrt{\frac{p^*(1-p^*)}{10^4}} \leq 1.96\sqrt{\frac{\frac{1}{2}(1-\frac{1}{2})}{10^4}}$$

$$\leq 0.01$$

Therefore the confidence limits on $P_{F/IA}$ and $P_{F/IB}$, are closer than ± 0.01.

In Tables 5.06-5.09, the estimates of the probabilities $P_{F/I}$, have been obtained by taking the average of the estimates for $P_{F/IA}$

and $P_{F/IB}$. Hence the confidence limits for the estimates of $P_{F/I}$ should be closer than those derived for $P_{F/IA}$ and $P_{F/IB}$.

5.09 Probability of System 1 Eventually Reaching the Failure Mode, Given the Appearance of a Merged Pair of Vectors

From the results given in Section 5.07, it is clear that the probabilities $P_{F/I}$, decrease fairly rapidly as I increases from 1 to 3. These results suggest that

$$P_{F/I} < 0.02$$

for I = 3, 4, 5, ..., k-1, for cases where the rate of errors is 0.004, or less. (i.e. where the noise level is such that the error rate is ≤ 0.004). k is, of course, the number of vectors stored at the start of each cycle of the detection process. It will now be assumed that the probabilities $P_{F/I}$ are negligibly small, for $I \geq 4$, where they are defined for such values of I.

Now let p_I be the probability that System 1 has entered mode I, given that it has entered either mode 1, 2 or 3. Also let P_F be the probability of the process eventually entering the failure mode, given that it has entered one of these three modes at some stage. Then

P_F = Prob. (the process had entered mode 1 and it eventually
enters mode F
or
the process had entered mode 2 and it eventually
enters mode F
or
the process had entered mode 3 and it eventually
enters mode F).

(Note that the probability of System 1 moving from mode I to the failure mode, has been assumed to be negligible for $I \geq 4$). Hence it may be seen that

$$P_F = \sum_{I=1}^{3} \text{Prob. (the process had entered mode I and it eventually enters mode F)}$$

$$= \sum_{I=1}^{3} \text{Prob. (the process had entered mode I)} \times\text{Prob.(the process eventually enters mode F, given that it had entered mode I)}$$

$$\therefore \quad P_F = \sum_{I=1}^{3} p_I \, P_{F/I} \qquad (5.36)$$

It has been shown in Section 5.02, that the occurrence of a pair of vectors which are merged with a small separation, is a fairly rare event. Hence it is clear that some very long simulation tests may be required, to estimate the probabilities p_I. In the absence of the necessary simulation results, certain values will now be assumed for p_1, p_2 and p_3, so that the above expression may be evaluated.

p_I is defined to be the probability that the process has entered mode I, given that it has entered either mode 1, 2 or 3. i.e. given that System 1 is in one of these three modes, but the particular mode entered is not known. (I may take the values 1, 2 and 3). It will now be assumed that p_I is independent of I, and of the number k of vectors stored at the start of each cycle of the process. It then follows that

$$p_1 = p_2 = p_3 = \frac{1}{3} \, .$$

Then, from equation 5.36,

$$P_F = \frac{1}{3} \sum_{I=1}^{3} P_{F/I}$$

It can be seen from the simulation results given in Section 5.07, that the probabilities $P_{F/I}$ do not vary greatly with the value of k. Hence, under the assumptions made above, it can be seen that P_F does not vary, to a large extent, with the value of k. i.e. the probability of System 1 eventually entering the failure mode, given that it has entered either mode 1, 2 or 3, at some stage, does not vary greatly with the value of k.

In Section 5.07 simulation tests are described in which System 1 was placed in either mode 1, 2 or 3 at some stage, and allowed to continue until it settled in mode R or mode F. With the tests for which k=4, it was found that the process always entered either mode R or mode F, within twenty cycles of the initial merging. However with k=16, many instances were observed in which the process required several thousand cycles to settle in one of these two modes. Hence, in cases where the detection process is used with fairly short data sequences, it is more likely to enter the failure mode if a small value of k is used.

This concludes the analysis of the merging phenomenon, on System 1, but some methods will now be discussed for preventing the occurrence of merged vectors.

5.10 Prevention of Merging

From Section 5.04, it may be seen that the presence of two vectors in System 1, which are merged with a small separation, may lead to the process entering the failure mode. i.e. the process may enter a mode of operation, in which the stored vectors form into merged pairs with some small separation. It is, however, possible to modify System 1 in such a way that vectors merged with a small separation, cannot be formed. This can be achieved by preventing the costs associated with the vectors, from becoming too close together.

The decision rule for System 1 is such that, in each cycle, the k vectors with smallest costs are selected from a set of mk vectors. (See Section 3.04). Now consider an implementation of decision rule 1, such that the selected vectors are ordered according to the size of their costs. (This was the implementation used for the simulation tests). Let the costs for the k selected vectors be denoted $c_1, c_2, \ldots\ldots, c_k$
where

$$c_i \leq c_{i+1} \qquad \text{for} \qquad i = 1, 2, \ldots\ldots, k-1.$$

Consider the following sequence of operations performed on these costs, for some given parameter α:

If $|c_2 - c_1| < \alpha$, set $c_2 = c_2 + \alpha$ (i.e. replace the value of c_2 by $c_2 + \alpha$).

If $|c_3 - c_2| < \alpha$, set $c_3 = c_3 + \alpha$ (where c_2 is the cost which has possibly been updated in the previous step).

If $|c_4 - c_3| < \alpha$, set $c_4 = c_4 + \alpha$

.

.

.

If $|c_k - c_{k-1}| < \alpha$, set $c_k = c_k + \alpha$.

In each step where a cost c_{i+1} is being compared with c_i, the value of c_i considered is the possible updated value, from the previous step.

Definition

System 1B is defined to be the version of System 1, which has the modification described above.

By means of this modification to the detection process, it is ensured that each cost is separated by an amount greater than or equal to α, from the other costs. Hence no pair of vectors may be present in the system, which are merged with a separation less than α. This modification does, of course, distort the costs for the stored vectors. It is hoped though, that a small value of α will prevent the system from entering the failure mode, while keeping this distortion of costs to a low level. It should be noted that System 1B is identical to System 1, if $\alpha = 0$.

Table 5.10 shows a section of computer output, from a simulation test on System 1, in which the failure mode occurred. The detection process was being tested with a two level signal, channel E, $\sigma = 0.0763$ and k=4 (where σ is the noise standard deviation, and k is the number of vectors stored at the start of each cycle of the process). This simulation test is the one in which the most dramatic drop in performance was observed, for System 1.

Number of cycles	Number of errors	Proportion of errors x 10^4
150000	28	1.87
2000	28	1.84
4000	28	1.82
6000	28	1.79
8000	28	1.77
160000	28	1.75
2000	28	1.73
4000	28	1.71
6000	28	1.69
8000	41	2.44
170000	43	2.53
2000	65	3.78
4000	109	6.26
6000	109	6.19
8000	188	10.56
180000	236	13.11
2000	261	14.34
4000	305	16.58
6000	316	16.99
8000	350	18.62
190000	350	18.42
2000	383	19.95
4000	387	19.95
6000	417	21.28
8000	432	21.82

Block of 2000 cycles	Number of errors in the block
22,000 - 24,000	7
28,000 - 32,000	2
58,000 - 60,000	13
100,000 - 102,000	2
120,000 - 122,000	2
146,000 - 148,000	2

TABLE 5.10

Computer output for System 1, channel E, k=4, N=11 and σ = 0.0763.

Number of cycles	Number of errors	Proportion of errors x 10^4
150000	28	1.87
2000	28	1.84
4000	28	1.82
6000	28	1.79
8000	28	1.77
160000	28	1.75
2000	28	1.73
4000	28	1.71
6000	28	1.69
8000	41	2.44
170000	43	2.53
2000	65	3.78
4000	109	6.26
6000	109	6.19
8000	188	10.56
180000	236	13.11
2000	261	14.34
4000	305	16.58
6000	305	16.40
8000	305	16.22
190000	305	16.05
2000	305	15.89
4000	305	15.72
6000	305	15.56
8000	305	15.40

TABLE 5.11

Computer print-out for System 1B (α = 0.0001), channel E, k=4, N=11 and σ = 0.0763.

Number of cycles	Number of errors	Proportion of errors x 10^4
150000	29	1.93
2000	29	1.91
4000	29	1.88
6000	29	1.86
8000	29	1.84
160000	29	1.81
2000	29	1.79
4000	29	1.77
6000	29	1.75
8000	42	2.50
170000	42	2.47
2000	44	2.56
4000	44	2.53
6000	44	2.50
8000	45	2.53
180000	45	2.50
2000	45	2.47
4000	45	2.45
6000	45	2.42
8000	45	2.39
190000	45	2.37
2000	45	2.34
4000	45	2.32
6000	45	2.30
8000	45	2.27

TABLE 5.12

Computer print-out for System 1B (α = 0.001), channel E, k=4, N=11 and σ = 0.0763

Number of cycles	Number of errors	Proportion of errors x 10^4
150000	20	1.33
2000	20	1.32
4000	20	1.30
6000	20	1.28
8000	20	1.27
160000	20	1.25
2000	20	1.23
4000	20	1.22
6000	20	1.20
8000	20	1.19
170000	20	1.18
2000	20	1.16
4000	20	1.15
6000	20	1.14
8000	22	1.24
180000	22	1.22
2000	22	1.21
4000	22	1.20
6000	22	1.18
8000	22	1.17
190000	22	1.16
2000	22	1.15
4000	22	1.13
6000	22	1.12
8000	22	1.11

TABLE 5.13

Computer print-out for System 1B (α = 0.01), channel E, k=4, N=11 and σ = 0.0763

Number of cycles	Number of errors	Proportion of errors x 10^4
150000	217	14.47
2000	226	14.87
4000	234	15.19
6000	243	15.58
8000	243	15.38
160000	243	15.19
2000	243	15.00
4000	253	15.43
6000	254	15.30
8000	269	16.01
170000	269	15.82
2000	269	15.64
4000	275	15.80
6000	289	16.42
8000	297	16.69
180000	304	16.89
2000	306	16.81
4000	306	16.63
6000	329	17.69
8000	331	17.60
190000	341	17.95
2000	343	17.86
4000	343	17.68
6000	353	18.01
8000	353	17.83

TABLE 5.14

Computer print-out for System 1B ($\alpha = 0.025$), channel E, k=4, N=11 and $\sigma = 0.0763$

Number of cycles	Number of errors	Proportion of errors x 10^4
150000	20867	1391.13
2000	21164	1392.37
4000	21488	1395.32
6000	21727	1392.76
8000	22034	1394.56
160000	22284	1392.75
2000	22489	1388.21
4000	22811	1390.91
6000	23118	1392.65
8000	23430	1394.64
170000	23678	1392.82
2000	23940	1391.86
4000	24290	1395.98
6000	24585	1396.88
8000	24834	1395.17
180000	25137	1396.50
2000	25394	1395.27
4000	25645	1393.75
6000	25976	1396.56
8000	26224	1394.89
190000	26461	1392.68
2000	26741	1392.76
4000	26973	1390.36
6000	27258	1390.71
8000	27579	1392.88

TABLE 5.15

Computer print-out for System 1B (α = 0.1) channel E, k=4, N=11 and σ = 0.0763

The computer output given in Table 5.10, shows the number of errors which had occurred to date, in the detected data sequence, at different stages of the test. It can be seen that, during the detection of the first 166,000 elements, only twenty-eight errors occurred. The distribution of the errors in this part of the simulation test, is also given in the table. It is clear from Table 5.10, that the detection process had deteriorated considerably after the first 166,000 cycles. (i.e. after the first 166,000 data elements had been detected). The number of errors occurring in the following 32,000 elements, was 404.

The four vectors stored by the detection process at the start of each cycle, were printed out at intervals of 50,000 cycles. It was found that these vectors were distinct after 50,000, 100,000 and 150,000 cycles. However, after 200,000 cycles (after the point at which the error rate had increased dramatically), these four vectors were found to be formed into two pairs of vectors, which were merged with a separation of 2.3 x 10^{-5}, i.e. the detection process had entered the failure mode, with separation $\varepsilon = 2.3 \times 10^{-5}$. These results suggest that the occurrence of the failure mode, was the reason for the sudden drop in the performance of the process.

Table 5.11 shows a section of the results from a simulation test on System 1B, in which the data and noise sequences used, were identical to those for the test performed on System 1. The two simulation tests were the same in every respect, apart from the modification which converts System 1 to System 1B. The value of α used for the test on System 1B, was 0.0001. It is clear from the description of System 1B, that the system is equivalent to System 1, if α is set equal to zero. It was hoped, therefore, that with a small value of α, the two systems would behave in a very similar manner, apart from situations where System 1 suffered from merging.

The simulation test with System 1B (and $\alpha = 0.0001$), revealed that it did produce the same error distribution as System 1, up to the point where System 1's performance was suddenly reduced. i.e. up to and including the 166,000 th. cycle of the process. A comparison of Tables 5.10 and 5.11 shows that the distribution of errors, for the two systems, was also identical up to the 184,000 th. cycle. At this stage, however, the performance of System 1B returned to that which would normally be expected for System 1. System 1 continued with a poor performance until the end of the simulation test, at the 250,000 th. cycle. The four vectors stored by System 1B, at the end of the 200,000 cycle, were found to be distinct whereas the vectors of System 1 formed two merged pairs. It appears therefore, that the two systems suffered from the merging phenomenon during cycles 166,000 to 184,000. System 1B (with $\alpha = 0.0001$) then seems to have escaped from the failure mode, while System 1 continued in this mode at least until the 250,000 th. cycle.

Further simulation tests were carried out on System 1B, which were identical to the one described above, apart from the fact that different values of α were used. Sections of the results for these tests are given in Tables 5.12-5.15. With $\alpha = 0.001$ (Table 5.12), it can be seen that System 1B had one more error in its detected data sequence, up to the 166,000 th. cycle. However, in the detection of the following elements, System 1B showed no noticeable drop in performance, whereas the error rate suddenly increased with System 1.

Tables 5.13-5.15 show the results of the tests on System 1B, for $\alpha = 0.01$, 0.025 and 0.1 respectively. It can be seen from these results, that $\alpha = 0.01$ gives the lowest over all error rate, over the five values of α which were used in the test. The performance of System 1B becomes very poor if α is increased to 0.1.

Clearly, for the situation examined in which System 1 had entered the failure mode, System 1B can offer an improved performance. It appears to be possible to find a value of α, for System 1B, which makes the system immune to the problem of merging, and which allows the performance normally expected of System 1.

A second modification to System 1 will now be discussed, with which the problems due to merging vectors may be eliminated.

During the j+2 nd. cycle of the System 1 detection process, the k vectors

$$\underline{Q}_j(1),\ \underline{Q}_j(2),\ \ldots\ldots,\ \underline{Q}_j(k)$$

are extended to mk vectors of the form

$$\underline{I}_{j+1}(I,J) = (\underline{Q}_j(I),J)$$

for $I = 1, 2, \ldots\ldots, k$

and $J = -m+1, -m+3, \ldots\ldots, m-1$.

(see Section 3.02), where m is the number of signal levels. From this set of mk extended vectors, the one with smallest cost is selected, and the element furthest to the left of this vector, is taken as a detected element. The k vectors of the form

$$\underline{I}_{j+1}\ (I,J)$$

with smallest costs, are then retained for the next cycle of the process.

Definition

Consider the earliest element (the element furthest to the left), of the k selected vectors of the form

$$\underline{T}_{j+1}(I,J)$$

Let System 1 be modified, so that any of these k vectors whose earliest element is not the same as the detected element, are removed from the process, during each cycle. Then System 1C is defined to be this modified version of System 1. Note that, if some of the k selected vectors are removed from the system, the following cycle of the process will commence with some number k_1 of stored vectors where $k_1 < k$. These k_1 vectors are then extended, in the usual way, to mk_1 vectors (where mk_1 is hopefully $\geq k$), and the k vectors with smallest costs are selected as before. If $mk_1 < k$, all of the mk_1 vectors are selected. This modified version of System 1 has previously been proposed by Vermeulen, for the case of $k = 2$ [40].

Now consider a situation in which System 1 enters the failure mode. It can be seen from the discussion at the end of Section 5.04, that the process will then go on to become locked in a state, in which there are only $\frac{1}{2}k$ distinct vectors. Simulation results confirm that the performance of the detector is sometimes reduced when this mode of operation occurs. It will be seen from the following theorem, that System 1C will always have a distinct set of stored vectors. It should not therefore have this weakness exhibited by System 1.

Theorem 5.02

Let System 1C be operated with the recommended starting up procedure, described in Section 3.08. Then, after the first few cycles of the process, the system will have a set of distinct stored vectors at the start of each cycle.

Proof

First consider a situation where the k vectors stored by System 1C, at the start of some cycle j, form a distinct set. Let the vectors be denoted

$$\underline{v}_1, \underline{v}_2, \ldots\ldots, \underline{v}_k.$$

Then, from the definition of System 1C, these vectors must be such that the same first component is common to each of them.

During the j th cycle of the process, the above k vectors are extended to mk vectors of the form

$$(\underline{v}_i, J)$$

where $i = 1, 2, \ldots., k$

and $J = -m+1, -m+3, \ldots\ldots, m-1.$

k of the mk vectors are then selected, and the N components furthest to the right of these vectors, are retained for the next cycle of the process.

Now suppose that two of the vectors, present at the start of the j+1 st cycle, are identical. Then there must exist two N+1 component vectors of the form

$$(\underline{v}_\ell, I_1) \text{ and } (\underline{v}_m, I_2)$$

which have their latest N components in common (i.e. the N components furthest to the right are the same for both vectors). All of the vectors

$$\underline{v}_1, \underline{v}_2, \ldots\ldots, \underline{v}_k$$

have their first components in common, so

$$(\underline{v}_\ell, I_1) = (\underline{v}_m, I_2)$$

and

$$\underline{v}_\ell = \underline{v}_m.$$

This is a contradiction, as the vectors

$$\underline{v}_1, \underline{v}_2, \ldots\ldots, \underline{v}_k$$

form a distinct set, hence the above supposition must be incorrect. i.e. it is not possible for two of the vectors, present at the start of the j+1st cycle, to be identical.

It has been shown above that, if the vectors of System 1C are distinct, at the start of one cycle of the process, this situation will be maintained at the start of the following cycle. Now refer to the proof of theorem 3.02 (Section 3.10). In the analysis of System A, i th cycle, it was shown that System 1 will have a set of k distinct vectors in store, at the end of the i th cycle of the process. (i is defined to be the smallest integer, such that $m^i \geq k$). Furthermore, the first component is common in each of the vectors, provided that the vectors have at least i+1 components. It can be seen that the analysis given for System 1, also applies to System 1C. Hence System 1C will have a set of distinct stored vectors at the start of the i+1st cycle. It follows, therefore, that the vectors must be distinct at the beginning of the i+2nd cycle, and this situation must be maintained for each of the following cycles, (applying an inductive argument). It has been assumed here that the recommended starting up procedure, described in Section 3.08, has been used with System 1C.

End of proof of theorem 5.02.

In this section, of the thesis, two detection processes, 'Systems 1B and 1C', have been proposed, which are modifications of System 1. For a particular situation considered, System 1B was able to overcome the loss in performance due to merging, while offering the same performance as System 1 where merging had not occurred.

Limited simulation tests have been carried out with System 1C. These tests suggest that its performance is generally as good as thatof System 1, if N is greater than about ten. System 1C should offer a definite improvement in tolerance to noise, over System 1, in situations where the failure mode occurs in the latter process. It is recommended that one of these two modified versions be used in preference to System 1, in any practical application.

CHAPTER 6

6.01 Originality

The work described in Chapters 3-5 of this thesis is believed to be original except where stated. The following are the more important contributions and are original, to the best of the author's knowledge:

i) All simulation results for Systems 1 and 2.

ii) The results of Section 3.08 which demonstrate the importance of the starting up procedure for Systems 1 and 2.

iii) The proof of the theorems of Section 3.10, which reveal the effect of an extra zero component at the start of a channel's sampled impulse response.

iv) The analysis of the merging phenomenon given in Chapter 5.

v) The modified algorithm, "System 1B", which is immune to merging.

6.02 Suggestions for Further Work

The work of this project, on Viterbi based detection processes, might usefully be extended along the lines of:

1. A comparison between Systems 1-4, and detection processes using a V.A. detector in conjunction with a linear or decision feedback equalizer. The latter types of detection processes are described and investigated in references 32, 36 and 44.

2. Simulation testing to determine the effect of quantizing all numbers stored by the algorithms. (The tests described in this thesis have been conducted with all numbers stored to a high degree of precision).

3. A study of the effect on performance, of ignoring some of the leading and trailing components of the channel's sampled impulse response, and thereby saving on storage and computation.

4. Simulation testing to study the effect of small errors in the channel's sampled impulse response, as estimated by the receiver.

5. Designing an adaptive process which adjusts the number of initial sampled impulse response components, ignored by the detector.

   It has been shown in Chapters 3 and 4, that the tolerance to noise of Systems 1-4 can sometimes be improved, if these detectors ignore some of the leading sampled impulse response components. With a time varying channel, the number n_γ of these components ignored, could be varied to suit changes in the transmission channel. An automatic process would then be needed to adjust n_γ when required, to obtain the best performance from the detectors.

6.03 Conclusions

This research project has been concerned with the study of detection processes which can offer a close to optimum performance, without the need for excessive computation. Under certain conditions, the optimum detection process is given by the Viterbi Algorithm (V.A) detector. Simulation tests have shown that this detector offers a considerable increase in tolerance to additive white Gaussian noise, over the conventional non linear equalizer, with channels which introduce severe amplitude distortion. However, for many typical situations, the computational demands of the V.A. detector render it impracticable.

Four detection processes, Systems 1-4, have been studied which are based on the V.A.. Unlike the V.A. detector, certain parameters may be varied in these systems to give the desired compromise between performance and complexity. For many situations it has been found that Systems 1-4 can offer a tolerance to noise which is quite close to that of an optimum detection process, with only a small fraction of the computation required by the V.A. detector.

Systems 1 and 2 have been found to suffer occasionally, from an effect called merging which can drastically reduce their performances. These detectors may however be modified to forms which do not suffer from merging.

In some applications, the impulse response of the transmission channel under consideration may grow slowly with time, so that the first few components of the sampled impulse response are small. Simulation tests show that it is sometimes of advantage with Systems 1-4, to ignore some of these small components. If these components are not ignored the complexity of the detectors may have to be increased, to obtain a given tolerance to noise.

It has been found that Systems 3 and 4 have weaknesses not exhibited by Systems 1 and 2. The performance of System 4 is not quite as good as that of Systems 1-3, over some channels with severe amplitude distortion. With System 3 a much greater loss in performance is experienced, due to the presence of a small component at the start of the channel's sampled impulse response, than with Systems 1, 2 and 4. The performances of Systems 1 and 2 are usually about the same, but System 2 requires the least number of basic operations of the two systems, per detected data element. System 2 therefore appears to be the most promising of the four detection processes, as a possible replacement for the conventional non linear equalizer.

APPENDIX 1

RESULTS CONCERNING CONDITIONAL PROBABILITY DENSITY FUNCTIONS (CONDITIONAL pdf's)

Let

$$\underline{s} = (s_0, s_1, \ldots\ldots, s_n)$$

be a vector whose components are discrete random variables and let

$$\underline{r} = (r_0, r_1, \ldots\ldots, r_{n+g}) \quad .$$

be a vector whose components are continuous random variables.

Let

$$G(\underline{r}', \underline{s}') = \text{Prob.}(\underline{r} \leq \underline{r}' \text{ and } \underline{s} = \underline{s}')$$

where $\underline{r} \leq \underline{r}'$ means that each component of $\underline{r}$ is $\leq$ the corresponding component of $\underline{r}'$. Then the joint pdf of $\underline{r}$ and $\underline{s}$ is defined by

$$g_4\ (\underline{r}', \underline{s}') = \frac{d}{dr_0'}\ \frac{d}{dr_1'} \cdots \frac{d}{dr_{n+g}'}\ G(\underline{r}', \underline{s}')$$

The conditional pdf of $\underline{r}$, when it is given that $\underline{s} = \underline{s}'$, is defined by

$$f(\underline{r}'/\underline{s}') = \frac{g_4\ (\underline{r}', \underline{s}')}{g_3\ (\underline{s}')} \qquad (1)$$

where $\quad g_3(\underline{s}') = \text{Prob.}(\underline{s} = \underline{s}')$.

The probability that $\underline{s} = \underline{s}'$, when it is given that $\underline{r} = \underline{r}'$, is defined by

$$g_1(\underline{s}'/\underline{r}') = \frac{g_4(\underline{r}', \underline{s}')}{g_2(\underline{r}')} \qquad (2)$$

where $g_2(\underline{r}')$ = pdf of $\underline{r}'$ or the joint pdf of r_0', r_1', ..., r_{n+g}'.

From equations 1 and 2, it is clear that

$$f(\underline{r}'/\underline{s}')\ g_3(\underline{s}') = g_1(\underline{s}'/\underline{r}')\ g_2(\underline{r}')$$

APPENDIX 2

RELATIONSHIP BETWEEN A CHANNEL'S SAMPLED IMPULSE RESPONSE AND ITS FOURIER TRANSFORM

Consider a transmission channel with impulse response y(t). Let Y(f) be the Fourier transform of y(t), so that

$$Y(f) = \int_{-\infty}^{\infty} y(t)\, e^{-jft.2\Pi}\, dt \qquad (1)$$

where f is the frequency in Hertz

and $j = \sqrt{-1}$.

Assume that the channel is band limited, so that

$$Y(f) = 0 \text{ when } |f| > B \text{ Hz}$$

for some value B. Also assume that Y(f) has a Fourier series expansion, so that

$$Y(f) = \sum_{i=-\infty}^{\infty} c_i\, e^{ijf\Pi/B} \qquad (2)$$

for $|f| < B$

where

$$c_i = \frac{1}{2B} \int_{-B}^{B} Y(f)\, e^{-\Pi ifj/B}\, df \qquad (3)$$

Note that y(t) is related to Y(f) by the inverse Fourier transform and

$$y(t) = \int_{-\infty}^{\infty} Y(f)\, e^{2\Pi jft}\, df$$

Hence

$$y(\tfrac{-i}{2B}) = \int_{-B}^{B} Y(f)e^{-\Pi jfi/B}\, df \qquad \text{(noting that } Y(f) = 0 \text{ for } |f| > B)$$

$\therefore$ from equation 3,

$$c_i \doteq \frac{1}{2B}\, y(\tfrac{-i}{2B})$$

Then, using this expression in equation 2 gives

$$Y(f) = \frac{1}{2B} \sum_{i=-\infty}^{\infty} y(\tfrac{-i}{2B})\; e^{\Pi ifj/B}$$

$$Y(f) = \frac{1}{2B} \sum_{i=-\infty}^{\infty} y(\tfrac{i}{2B})\; e^{-\Pi ifj/B} \qquad \text{for } |f| < B \qquad (4)$$

Now suppose that $y(\frac{i}{2B})$ can be considered to be negligibly small except for $i = 0, 1, 2, \ldots. g$, for some non-negative integer g. Also assume that

$$y(\tfrac{0}{2B}),\ y(\tfrac{1}{2B}),\ \ldots\ldots,\ y(\tfrac{g}{2B})$$

forms the sampled impulse response of the channel, so that the impulse response is sampled at intervals of $\frac{1}{2B}$. (This sampling rate is called the Nyquist rate for the channel). Then equation 4 gives

$$Y(f) = \frac{1}{2B} \sum_{i=0}^{g} y(\tfrac{i}{2B})\; e^{-\Pi ifj/B}$$

$$= \frac{1}{2B} \sum_{i=0}^{g} y_i\; e^{-\Pi ifj/B} \qquad \text{for } |f| < B$$

where $y_i = y(\frac{i}{2B})$

for $i = 0, 1, \ldots\ldots, g.$

Hence, with the assumptions made above, the Fourier transform of the channel's impulse response is given, within a constant multiple, by

$$Y(f) = \sum_{i=0}^{g} y_i \, e^{-\Pi i f j/B} \qquad \text{for } |f| < B$$

where $(y_0, y_1, \ldots\ldots, y_g)$

is the channel's sampled impulse response.

APPENDIX 3

COMPUTER PROGRAMS

Computer programs for Systems 1 and 4 are listed in this appendix. It should be noted that System 4 is identical to the Viterbi Algorithm, if used with $k = m^g$, i.e. if used with m^g vectors stored at the start of each cycle.

The two programs were written in 1900 Fortran, which is very similar to Fortran IV. Both programs make use of three routines from the Numerical Algorithms Group (NAG) Library. These routines are:

```
G05BAF (T)
G05AAF (FX)
G05AEF (0.0,S)
```

They are each concerned with the generation of pseudo random numbers.

G05BAF (T) initializes the basic random number generator, which the other two routines use to form their number sequences. T is a parameter supplied by the user, which controls the starting point in the random number sequence. T must lie in the range [0,1].

G05AAF (FX) supplies a pseudo random number from a uniform [0,1] distribution. FX is a dummy parameter.

G05AEF (0.0,S) supplies a pseudo random number from a normal distribution with mean 0.0 and standard deviation S. The value of S is supplied by the user.

```
      MASTER VITER
      DIMENSION IS(15,32),B1(32),C1(32,2),A1(2),A2(2),A3(2),A4(2),A5(2
     1 A6(2),A7(2),A8(2),A9(2),A10(2),A11(2),A12(2),A13(2),A14(2),A15(
     2 ,JR(20),J1(32),K1(32)
      COMMON JV1,JV2,JV3,JV4,JV5,JV6,JV7,JV8,JV9,JV10,JV11,JV12,JV13,
     1 JV14,M1,N1,JR,LIM,S
C  SYSTEM 1 WITH IN STORED VECTORS AND A BINARY SIGNAL
      READ(1,908)A1(1),A2(1),A3(1),A4(1),A5(1)
      READ(1,908)A6(1),A7(1),A8(1),A9(1),A10(1)
      READ(1,908)A11(1),A12(1),A13(1),A14(1),A15(1)
C  A1(1),A2(1),...,A15(1) IS THE SAMPLED IMPULSE RESPONSE OF THE
C... CHANNEL BEING TESTED
 4    READ(1,906)S
C  S IS THE NOISE STANDARD DEVIATION
      WRITE(2,902)
      WRITE(2,903)A1(1),A2(1),A3(1),A4(1),A5(1)
      WRITE(2,903)A6(1),A7(1),A8(1),A9(1),A10(1)
      WRITE(2,903)A11(1),A12(1),A13(1),A14(1),A15(1)
      WRITE(2,904)S
      A1(2)=2.0*A1(1)
      A2(2)=2.0*A2(1)
      A3(2)=2.0*A3(1)
      A4(2)=2.0*A4(1)
      A5(2)=2.0*A5(1)
      A6(2)=2.0*A6(1)
      A7(2)=2.0*A7(1)
      A8(2)=2.0*A8(1)
      A9(2)=2.0*A9(1)

      A10(2)=2.0*A10(1)
      A11(2)=2.0*A11(1)
      A12(2)=2.0*A12(1)
      A13(2)=2.0*A13(1)
      A14(2)=2.0*A14(1)
      A15(2)=2.0*A15(1)
C  ALL POSSIBLE  VALUES THE TERM I*A1(J) ARE CALCULATED ,FOR
C...  I=1,2  AND J=1,2,...,15
C  THE POSSIBLE VALUES OF A DATA ELEMENT ARE 1 AND 2 IN THIS PROGRAM
      READ(1,912) IN
      WRITE(2,913) IN
C  IN IS THE NUMBER OF VECTORS STORED AT THE START OF EACH CYCLE OF
C... THE PROCESS
      IE=0
      IC2=20001
      WRITE(2,905)
      WRITE(2,907)
      WRITE(2,909)
      IC=-10
      ICOUNT=-10
      T=0.028
      CALL G05BAF(T)
      CALL G05BAF(T)
      LIM=12
      LIM1=LIM+1
      DO 2 I=1,LIM1
      JR(I)=1
 2    CONTINUE
C  SOME OF THE MOST RECENTLY TRANSMITTED DATA ELEMENTS ARE STORED IN
C... THE ARRAY JR(.)
      M1=2
      N1=LIM+2
      READ(1,901)JV1,JV2,JV3,JV4,JV5,JV6,JV7,JV8,JV9,JV10,JV11,JV12,JV
     1 ,JV14
      DO 9 I=1,15
      DO 11 J=1,32
      IS(I,J)=1
 11   CONTINUE
 9    CONTINUE
C  IS(I,J)=COMPONENT I OF THE J TH, STORED VECTOR
      B1(1)=0.0
      IF(IN .EQ. 1)GO TO 140
      DO 10 J=2,IN
      B1(J)=1.0E06
 10   CONTINUE
C  B1(J)=COST FOR THR STORED VECTOR :
C  [IS(1,J),IS(2,J),....,IS(15,J)]
 140  CALL GENERA(Z,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12,A13,A14,A15
C  SUBROUTINE GENERA SUPPLIES A NEW RECEIVED SIGNAL SAMPLE Z,EACH TIME
C... IT IS CALLED
      DO 80 J=1,IN
      N02=IS(2,J)
      N03=IS(3,J)
      N04=IS(4,J)
      N05=IS(5,J)
      N06=IS(6,J)
      N07=IS(7,J)
      N08=IS(8,J)
      N09=IS(9,J)
      N10=IS(10,J)
      N11=IS(11,J)
      N12=IS(12,J)


      N13=IS(13,J)
      N14=IS(14,J)
      N15=IS(15,J)
      B=A2(N15)+A3(N14)+A4(N13)+A5(N12)+A6(N11)+A7(N10)+A8(N09)+A9(N08)
     1 A10(N07)+A11(N06)+A12(N05)+A13(N04)+A14(N03)+A15(N02)-Z
      B11=B1(J)
      DO 90 K=1,2
      A=B+A1(K)
      C1(J,K)=B11+A*A
 90   CONTINUE
 80   CONTINUE
C  C1(J,K)=COST FOR THE VECTOR :
C  [IS(1,J),IS(2,J),....,IS(15,J),K]
      CALL MIN(C1,J1,K1,IN)
C  SUBROUTINE MIN SELECTS THE IN PAIRS OF VALUES (J,K) WHICH GIVE THE
C... SMALLEST IN VALUES OF THE COST C(J,K)
```

```
      J2=J1(1)
      K2=K1(1)
      IND=1
      IF(C1(J2,K2)-1.0E06)6,6,7
 7    IND=2
 6    IF(IS(5,J2) .NE. JR(M1))IE=IE+1
C A DETECTION IS MADE FROM THE FIFTH COMPONENT OF THE VECTOR WITH
C... SMALLEST COST , THUS GIVING A DELAY OF 11 SAMPLING INTERVALS
C... BETWEEN TRANSMISSION AND DETECTION.
C  JR(M1) IS ONE OF THE DATA ELEMENTS TRANSMITTED EARLIER
C IE=NUMBER OF ERRORS
      IF(ICOUNT .NE. 500) GO TO 3
      ICOUNT=0
      A111=IE
      A112=IC
C IC=NUMBER OF TRANSMITTED ELEMENTS
      IF(IC .EQ. 0) A112=1.0
      RATE=A111/A112
      WRITE(2,900) IC,IE,RATE
 3    CONTINUE
      IC=IC+1
      ICOUNT=ICOUNT+1
      DO 210 J=1,14
      DO 220 I=1,IN
      J2=J1(I)
      IS(J,I)=IS(J+1,J2)
 220  CONTINUE
 210  CONTINUE
      DO 240 I=1,IN
      IS(15,I)=K1(I)
      J2=J1(I)
      K2=K1(I)
      B1(I)=C1(J2,K2)
      IF(IND-1)240,240,8
 8    B1(I)=B1(I)-1.0E06
 240  CONTINUE
C  IN THE ABOVE 14 STATEMENTS, THE SELECTED VECTORS AND THEIR COSTS AR
C... STORED IN THE LOCATIONS OF THE ORIGINAL VECTORS AND COSTS
      IF(IC-IC2)13,14,13
 14   WRITE(2,905)IS
      WRITE(2,909)B1
      IC2=IC2+150000
 13   CONTINUE
      IF(IC-20000) 140,140,1
 1    STOP
 900  FORMAT(1X,'IC,IE,RATE=',2X,I7,2X,I7,2X,F12.9)
 901  FORMAT(14(1X,I1))
 902  FORMAT(1X,'SAMPLED IMPULSE RESPONSE COMPONENTS :')
 903  FORMAT(5(2X,F10.6))
 904  FORMAT(1X,'NOISE STANDARD DEVIATION =',F10.6)
 905  FORMAT(1X,'IC IS THE NUMBER OF DATA ELEMENTS  DETECTED SO FAR')
 906  FORMAT(F10.6)
 907  FORMAT(1X,'IE IS THE NUMBER OF ERRORS SO FAR ,IN THE DETECTED '/
     1 'DATA SEQUENCE')
 908  FORMAT(5F10.6)
 909  FORMAT(1X,'RATE IS THE ERROR RATE OR PROPORTION OF ERRORS')
 912  FORMAT(I2)
 913  FORMAT(1X,'NUMBER OF VECTORS STORED AT THE START OF EACH CYCLE='
     1  I2)
      END
      SUBROUTINE MIN(C1,J1,K1,IN)
C  THIS SUBROUTINE SELECTS THE IN PAIRS OF VALUES (J,K) WHICH GIVE
C... THE IN SMALLEST VALUES OF THE COST C(J,K),THESE VALUES ARE DENOTE
C... (J1(1),K1(1)),(J1(2),K1(2)) ,...,(J1(IN),K1(IN))
      DIMENSION C1(32,2),J1(32),K1(32),IAB(32,2)
      DO 60 I=1,IN
      DO 70 J=1,2
      IAB(I,J)=1
 70   CONTINUE
 60   CONTINUE
      DO 30 K=1,IN
      AM= 1.0E08
      I2=1
      J2=1
      DO 40 I=1,IN
      DO 50 J=1,2
      IF(IAB(I,J))50,50,10
 10   IF(C1(I,J)-AM)20,50,50
 20   AM=C1(I,J)
      I2=I
      J2=J
 50   CONTINUE
 40   CONTINUE
      J1(K)=I2
      K1(K)=J2
      IAB(I2,J2)=-1
 30   CONTINUE
      RETURN
      END
      SUBROUTINE GENERA(Z,A1,A2,A3,A4,A5,A6,A7,A8,A9,A10,A11,A12,A13,A
     1 ,A15)
C  THIS SUBROUTINE SUPPLIES A RECEIVED SIGNAL SAMPLE Z EACH TIME IT
C... IS CALLED
      DIMENSION JR(20),A1(2),A2(2),A3(2),A4(2),A5(2),A6(2),A7(2),A8(2)
     1 A9(2),A10(2),A11(2),A12(2),A13(2),A14(2),A15(2)
      COMMON JV1,JV2,JV3,JV4,JV5,JV6,JV7,JV8,JV9,JV10,JV11,JV12,JV13,

     1 JV14,M1,N1,JR,LIM,S
      Y=G05AAF(FX)
      IX=2
      IF(Y-0.5)2,3,3
 2    IX=1
 3    JV15=IX
      Z=A1(JV15)+A2(JV14)+A3(JV13)+A4(JV12)+A5(JV11)+A6(JV10)+A7(JV9)+
     1 A8(JV8)+A9(JV7)+A10(JV6)+A11(JV5)+A12(JV4)+A13(JV3)+A14(JV2)+
     2 A15(JV1)
      X=G05AEF(0.0,S)
      Z=Z+X
      JV1=JV2
      JV2=JV3
      JV3=JV4
      JV4=JV5
      JV5=JV6
      JV6=JV7
      JV7=JV8
      JV8=JV9
      JV9=JV10
      JV10=JV11
      JV11=JV12
      JV12=JV13
      JV13=JV14
      JV14=JV15
      JR(N1)=IX
C  SOME OF THE MOST RECENTLY TRANSMITTED DATA ELEMENTS ARE STORED IN
C... THE ARRAY JR(.)
      M1=M1+1
      N1=N1+1
      IF(M1 .EQ. LIM+3)M1=1
      IF(N1 .EQ. LIM+3)N1=1
      RETURN
      END
 SAMPLED IMPULSE RESPONSE COMPONENTS :
    0.100000     0.166700     0.500000     0.666700     0.500000
    0.166700     0.000000     0.000000     0.000000     0.000000
    0.000000     0.000000     0.000000     0.000000     0.000000
 NOISE STANDARD DEVIATION =  0.094200
 NUMBER OF VECTORS STORED AT THE START OF EACH CYCLE= 4
 IC IS THE NUMBER OF DATA ELEMENTS  DETECTED SO FAR
 IE IS THE NUMBER OF ERRORS SO FAR ,IN THE DETECTED
DATA SEQUENCE
 RATE IS THE ERROR RATE OR PROPORTION OF ERRORS
 IC,IE,RATE=       500         14   0.028000000
 IC,IE,RATE=      1000         58   0.058000000
 IC,IE,RATE=      1500         66   0.044000000
 IC,IE,RATE=      2000         92   0.046000000
 IC,IE,RATE=      2500        122   0.048800000
 IC,IE,RATE=      3000        156   0.052000000
 IC,IE,RATE=      3500        169   0.048285714
 IC,IE,RATE=      4000        184   0.046000000
 IC,IE,RATE=      4500        238   0.052888889
 IC,IE,RATE=      5000        243   0.048600000
 IC,IE,RATE=      5500        274   0.049818182
 IC,IE,RATE=      6000        336   0.056000000
 IC,IE,RATE=      6500        359   0.055230769
 IC,IE,RATE=      7000        380   0.054285714
 IC,IE,RATE=      7500        400   0.053333333
 IC,IE,RATE=      8000        430   0.053750000
 IC,IE,RATE=      8500        464   0.054588235
 IC,IE,RATE=      9000        490   0.054444444
 IC,IE,RATE=      9500        513   0.054000000
 IC,IE,RATE=     10000        519   0.051900000
```

```
      MASTER VITER
C  SYSTEM 4
      DIMENSION B1(30),A1(30,16),IQ(30,64),JV1(30),U(64),V(64,16),
     1 ILK(64,16)
      CALL G05BAF(0.0)
      CALL G05BAF(0.0)
      IG=2
      IG1=IG+1
C  IG1 IS THE NUMBER OF SAMPLED IMPULSE RESPONSE COMPONENTS
      READ(1,908)(B1(I),I=1,IG1)
      WRITE(2,900)
      WRITE(2,902)(B1(I),I=1,IG1)
C  B1(1),B1(2),...,B1(IG1) IS THE SAMPLED IMPULSE RESPONSE OF THE
C... CHANNEL BEING TESTED
      READ(1,906)S
      WRITE(2,903)S
C  S IS THE NOISE STANDARD DEVIATION
      M=2
C  M=NUMBER OF SIGNAL LEVELS
      DO 1 J=1,M
      M1=-M+2*J-1
      DO 2 I=1,IG1
      A1(I,J)=B1(I)*M1
 2    CONTINUE
 1    CONTINUE
C  ALL POSSIBLE VALUES OF THE TERM B1(I)*M1 ARE STORED ,FOR
C... I=1,2,...,IG1 AND M1=-M+1,-M+3,...,M-1
      READ(1,912)IN,N
      WRITE(2,901) IN

      WRITE(2,904) N
      WRITE(2,905)
      WRITE(2,907)
      WRITE(2,909)
C  M**IN=NUMBER OF VECTORS STORED AT THE START OF EACH CYCLE OF THE
C... PROCESS
C  N=NUMBER OF COMPONENTS OF THE VECTORS STORED AT THE START OF A
C... CYCLE
      E=0.0
      IC=-N+1
      N3=N+1
      NI=N-IN
      M1=M**IN
      DO 3 L=1,M1
      AL=L
      DO 4 I=1,IN
      A=AL/(M**(IN-I))
      IA=A
      IF(A-IA .GT. 0.0)IA=IA+1
      IQ(N+1-I,L)=IA
      AL=AL-M**(IN-I)*(IA   -1)
 4    CONTINUE
      DO 5 I=1,NI
      IO(I,L)=1
 5    CONTINUE
 3    CONTINUE
```

```
C  IQ(I,J)=COMPONENT I OF THE J TH, STORED VECTOR
C  THE STORED VECTOR WITH LATEST IN COMPONENTS:J1,J2,...,JIN IS
C... DENOTED VECTOR L,WHERE L=J1+M*(J2-1)+M**2*(J3-1)+...
C...+M**(IN-1)*(JIN-1)
C  THE ABOVE 13 STATEMENTS ASSIGN INITIAL VALUES TO THE COMPONENTS
C... OF THE STORED VECTORS
      DO 6 I=1,N3
      JV1(I)=1
 6    CONTINUE
C  SOME OF THE MOST RECENTLY TRANSMITTED DATA ELEMENTS ARE STORED
C... IN THE ARRAY JV1(,)
      U(1)=0.0
      DO 7 L=2,M1
      U(L)=1.0E06
 7    CONTINUE
C  U(L) IS THE COST FOR THE STORED VECTOR : [IQ(1,L),IQ(2,L),,,,IQ(N,L
 140  CALL TRANS(Z1,A1,JV1,IG,M,S,N)
C  SUBROUTINE TRANS SUPPLIES A NEW RECEIVED SIGNAL SAMPLE Z1,EACH
C... TIME IT IS CALLED
      M2=M**(IN-1)
      DO 8 I=1,M
      DO 8 L=1,M2
      IL= I+M*(L-1)
      A=U(IL)
      DO 8 K=1,M
      B=A1(1,K)
      DO 9 J=1,IG
      IQJ=IQ(N-J+1,IL)
      B=B+A1(J+1,IQJ)
 9    CONTINUE
      B=Z1-B
      V(IL,K)=A+B*B
 8    CONTINUE
C  V(IL,K) IS THE COST FOR THE VECTOR:
C...  [IQ(1,IL),IQ(2,IL),...,IQ(N,IL),K]
      CALL MIN(V,M2,M,ILK,L1,K1)

C  SUBROUTINE MIN SELECTS A NUMBER M**IN OF VECTORS OF THE FORM:
C...  [IQ(1,L),IQ(2,L),...,IQ(N,L),K]  FOR USE IN THE NEXT CYCLE OF
C... THE PROCESS
      I=ILK(L1,K1)+M*(L1-1)
      IF(IQ(1,I) .NE. JV1(1))E=E+1.0
C  A DATA ELEMENT IS DETECTED FROM THE FIRST COMPONENT OF THE VECTOR
C... WITH SMALLEST COST
C  E IS THE NUMBER OF ERRORS SO FAR,IN THE DETECTED DATA SEQUENCE
C  JV1(1) IS ONE OF THE DATA ELEMENTS TRANSMITTED EARLIER
      IF(IC)13,13,32
 32   RATE=E/IC
 13   JC=IC/500
      C=IC/500.0
C  IC IS THE NUMBER OF DATA ELEMENTS TRANSMITTED SO FAR
      IF(C-JC)19,20,19
 20   WRITE(2,911) IC,E,RATE
 19   CONTINUE
      IC=IC+1
      N1=N-1
      DO 11 I=1,N1
      DO 10 L=1,M2
      DO 10 K=1,M
      LK=L+M2*(K-1)
      I1=ILK(L,K)+M*(L-1)
      IQ(I,LK)=IQ(I+1,I1)
 10   CONTINUE
 11   CONTINUE
      DO 12 L=1,M2
      DO 12 K=1,M
      LK=L+M2*(K-1)
      I1=ILK(L,K)+M*(L-1)
      IQ(N,LK)=K
      U(LK)=V(I1,K)
 12   CONTINUE
C  IN THE ABOVE 15 STATEMENTS,THE SELECTED VECTORS AND THEIR COSTS
C... ARE STORED IN THE LOCATIONS OF THE ORIGINAL VECTORS AND COSTS
      GO TO 140
 900  FORMAT(1X,'SAMPLED IMPULSE RESPONSE COMPONENTS;')
 901  FORMAT(1X,'NUMBER OF VECTORS STORED AT THE START OF EACH CYCLE',
     1  1X,'=M**',I2)
 902  FORMAT(5(2X,F10.6))
 903  FORMAT(1X,'NOISE STANDARD DEVIATION =',F10.6)
 904  FORMAT(1X,'NUMBER OF COMPONENTS OF THE VECTORS STORED AT THE'/
     1  1X,' START OF A CYCLE =',I2)
 905  FORMAT(1X,'IC IS THE NUMBER OF DATA ELEMENTS  DETECTED SO FAR')
 906  FORMAT(F10.6)
 907  FORMAT(1X,'IE IS THE NUMBER OF ERRORS SO FAR ,IN THE DETECTED ',
     1  1X,'DATA SEQUENCE')
 908  FORMAT(5F10.6)
 909  FORMAT(1X,'RATE IS THE ERROR RATE OR PROPORTION OF ERRORS')
 911  FORMAT(1X,'IC,E,RATE=',I6,2X,F7.0,2X,F8.6)
 912  FORMAT(I2,2X,I2)
      END
      SUBROUTINE TRANS(Z1,A,JV,IG,M,S,N)
C  THIS SUBROUTINE SUPPLIES A RECEIVED SIGNAL SAMPLE Z1,EACH TIME IT
C... IS CALLED
      DIMENSION A(30,16),JV(30)
      Y=G05AAF(FX)
      AM=M
      DO 10 I=1,M
      IF(Y .GE. (I-1)/AM .AND. Y .LE. I/AM)IX=I
 10   CONTINUE
      Y1=-M+2*IX-1
      JV (N+2)=IX
      Z1=0.0
      IG1=IG+1
      DO 20 I=1,IG1
      J=JV(N+3-I)
      Z1=Z1+A(I,J)
 20   CONTINUE
      X=G05AEF(0.0,S)
      Z1=Z1+X
      N3=N+1
      DO 30 I=1,N3
      JV(I)=JV(I+1)
 30   CONTINUE
C SOME OF THE MOST RECENTLY TRANSMITTED DATA ELEMENTS ARE STORED IN
C... THE ARRAY JV(.)
      RETURN
      END
      SUBROUTINE MIN(V,M2,M,ILK,L1,K1)
C THIS SUBROUTINE SELECTS A NUMBER M**IN OF VECTORS OF THE FORM:
C...  [IQ(1,L),IQ(2,L),...,IQ(N,L),K] ACCORDING TO DECISION RULE 4
      DIMENSION V(64,16),ILK(64,16)
      L1=0
      K1=0
      BM=1.0E06
      DO 1 L=1,M2
      DO 1 K=1,M
      I1=0
      AM=1.0E08
      DO 2 I=1,M
      A=V(I+M*(L-1),K)
      IF(A-AM)3,4,4
 3    AM=A
      I1=I
 4    CONTINUE
      IF(A-BM)5,6,6
 5    BM=A
      L1=L
      K1=K
 6    CONTINUE
 2    CONTINUE
      ILK(L,K)=I1
 1    CONTINUE
      RETURN
      END
```

```
SAMPLED IMPULSE RFSPONSE COMPONENTS:
   0.408000    0.816000    0.408000
NOISE STANDARD DEVIATION =  0.283000
NUMBER OF VECTORS STORED AT THE START OF EACH CYCLE
*M** 2
NUMPER OF COMPONENTS OF THE VECTORS STORED AT THE
 START OF A CYCLE =12
IC IS THE NUMBER OF DATA ELEMENTS  DETECTED SO FAR
IE IS THE NUMBER OF ERRORS SO FAR ,IN THE DETECTED
DATA SEQUENCE
RATE IS THE ERROR RATE OR PROPORTION OF ERRORS
IC,F,RATE=      0          0.  0.000000
IC,E,RATE=    500         10.  0.020000
IC,E,RATE=   1000         12.  0.012000
IC,E,RATE=   1500         16.  0.010667
IC,E,RATE=   2000         20.  0.010000
IC,E,RATE=   2500         22.  0.008800
IC,E,RATE=   3000         25.  0.008333
IC,E,RATE=   3500         30.  0.008571
IC,E,RATE=   4000         32.  0.008000
IC,E,RATE=   4500         37.  0.008222
IC,E,RATE=   5000         42.  0.008400
IC,E,RATE=   5500         42.  0.007636
IC,E,RATE=   6000         44.  0.007333
IC,E,RATE=   6500         46.  0.007077
IC,E,RATE=   7000         48.  0.006857
IC,E,RATE=   7500         52.  0.006933
IC,E,RATE=   8000         56.  0.007000
IC,E,RATE=   8500         59.  0.006941
```

APPENDIX 4

CONFIDENCE LIMITS FOR THE PROPORTION OF ERRORS

Consider a set of independent (Bernoulli) trials, with each trial having the two possible outcomes: success or failure. Let the probability of success in each trial be p, so that the probability of failure is 1-p. Let h be the proportion of successes in n such trials.

Then

$$\text{Prob.}\left(\frac{|h-p|}{\sqrt{h(1-h)/n}} \le K\right) \simeq 2\,\Phi(K) - 1$$

for any positive value of K, where

$$\Phi(K) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{K} e^{-t^2/2}\, dt$$

(see reference 50). Hence

$$\text{Prob.}\left(h-K\sqrt{\frac{h(1-h)}{n}} \le p \le h + K\sqrt{\frac{h(1-h)}{n}}\right) = 2\,\Phi(K) - 1$$

(It has been assumed here that n is large enough for the distribution of the random variable h, to be approximately normal). Now assume that h is small so that $\sqrt{1-h} \simeq 1$.

Then

$$\text{Prob.}\left(h-K\sqrt{\frac{h}{n}} \le p \le h + K\sqrt{\frac{h}{n}}\right) = 2\,\Phi(K) - 1$$

or

$$\text{Prob.}\left(h - \frac{Kh}{\sqrt{r}} \le p \le h + \frac{Kh}{\sqrt{r}}\right) = 2\,\Phi(K) - 1$$

where r is the number of errors occurring in the n trials, i.e. $r = nh$.

Now let K be such that

$$2\Phi(K) - 1 = 0.95$$

or

$$\Phi(K) = 0.975$$

where $\Phi(K)$ is defined above. From tables of the normal distribution function, $K = 1.96$.

Therefore

$$\text{Prob.}(h - 1.96\frac{h}{\sqrt{r}} \leq p \leq h + 1.96\frac{h}{\sqrt{r}}) = 0.95$$

REFERENCES

1. C E Shannon, *"A mathematical theory of communication"*, Bell System Technical Journal, Vol. XXVII, pp.379-423, 623-656, July, October 1948.

2. I Selvin, *"Detection theory"*, Princeton University Press, New Jersey, 1965.

3. D W Tufts, *"Nyquist problem - the joint optimization of transmitter and receiver in pulse amplitude modulation"*, Proc. IEEE, Vol. 53, pp.248-259, March 1965.

4. J W Smith, *"The joint optimization of transmitted signal and receiving filter for data transmission systems"*, Bell System Technical Journal, Vol. 44, pp.2363-2392, December 1965.

5. R W Lucky, J Salz and E J Weldon, Jr., *"Principles of data communication"*, McGraw Hill, New York, 1968.

6. A P Clark, *"Advanced data - transmission systems"*, Pentech Press, 1977.

7. B P Lathi, *"An introduction to random signals and communication theory"*, Intertext Books, 1970.

8. J C Hancock and P A Wintz, *"Signal detection theory"*, McGraw Hill, 1966.

9. A P Clark, *"Principles of data transmission"*, Pentech Press, 1976.

10. J M Wozencraft and I M Jacobs, *"Principles of communication engineering"*, Wiley, New York, 1965.

11. T H Crowley, G G Harris, S E Miller, J R Pierce and J P Runyon, *"Modern communications"*, Columbia University Press, 1962.

12. J L Lawson and G E Uhlenbeck, *"Threshold signals"*, Dover Publications Inc., New York, 1965.

13. H R Raemer, *"Statistical communication theory and applications"*, Prentice-Hall, 1969.

14. A P Clark and U S Tint, *"Linear and non linear transversal equalizers for baseband channels"*, The Radio and Electronic Engineer, Vol. 45, No. 6, pp.271-283, June 1975.

15. R W Lucky, *"Automatic equalization for digital communication"*, Bell System Technical Journal, Vol. 44, pp.547-588, April 1965.

16. R W Lucky, *"Techniques for adaptive equalization of digital communication systems"*, Bell System Technical Journal, Vol. 45, pp.255-286, February 1966.

17. E E Newhall, S U H Qureshi and C F Simone, *"A technique for finding approximate inverse systems and its application to equalization"*, IEEE Trans. on Communication Technology, Com-19, pp.1116-1127, December 1971.

18. J G Proakis, *"Adaptive digital filters for equalization of telephone channels"*, IEEE Trans. on Audio and Electronic Acoustics, AU-18, pp.195-200, 1970.

19. A P Clark, *"Adaptive detection of distorted digital signals"*, Radio and Electronic Engineer, Vol. 40, pp.107-119, September 1970.

20. P Monsen, *"Feedback equalization for fading dispersive channels"*, IEEE Trans. on Information Theory, IT-17, pp.56-64, January 1971.

21. D P Taylor, *"Non-linear feedback equalizer employing a soft limiter"*, Electronic Letters, Vol. 7, pp.265-267, 20 May 1971.

22. R T Boyd and F C Monds, *"Adaptive equalizer for multi-path channels"*, Electronic Letters, Vol. 6, pp.556-558, 20 August 1970.

23. D A George, R R Bowen and J R Storey, *"An adaptive feedback equalizer"*, IEEE Trans. on Communication Technology, Vol. COM-19, No. 3, pp.281-293, June 1971.

24. A P Clark, *"Design technique for non-linear equalizers"*, Proc. IEE, Vol. 120, pp.329-333, March 1973.

25. A Clements, *"The application of iterative techniques to adaptive detection processes"*, PhD Thesis, Loughborough University, 1976.

26. A P Clark and F Ghani, *"Detection processes for orthogonal groups of digital signals"*, IEEE Trans. on Communications, Vol. COM-21, pp.907-915, August 1973.

27. F Ghani, *"Serial digital communication systems with signals arranged in orthogonal groups"*, PhD Thesis, Loughborough University, 1974.

28. A P Clark, *"A synchronous serial digital data transmission system using orthogonal groups of binary signal elements"*, IEEE Transactions on Communication Technology, Vol. COM-19, Pt. I, pp.1101-1110, December 1971.

29. A P Clark, *"Adaptive detection with intersymbol interference cancellation for distorted digital signals"*, IEEE Trans. on Communications, pp.350-361, June 1972.

30. R A Gonsalves, *"Maximum likelihood receiver for digital data transmission"*, IEEE Trans. on Communications Technology, Vol. COM-16, pp.392-398, 1968.

31. A P Clark and J Harvey, *"Detection processes for distorted binary signals"*, The Radio and Electronic Engineer, Vol. 46, No. 11, November 1976.

32. D D Falconer and F R Magee, Jr., *"Adaptive channel memory truncation for maximum likelihood sequence estimation"*, Bell System Technical Journal, Vol. 52, No. 9, pp.1541-1563, November 1973.

33. G D Forney, Jr., *"The Viterbi Algorithm"*, IEEE Proc., Vol. 61, No. 3, March 1973.

34. A J Viterbi, *"Error bounds for convolutional codes and an asymptotically optimum decoding algorithm"*, IEEE Trans. on Information Theory, Vol. IT-13, pp.260-269, April 1967.

35. G D Forney, Jr., *"Maximum likelihood sequence estimation of digital sequences in the presence of intersymbol interference"*, IEEE Trans. on Information Theory, Vol. IT-18, pp.363-378, May 1972.

36. D D Falconer and F R Magee, Jr., *"Evaluation of decision feedback equalization and Viterbi detection for voiceband data transmission. Parts I and II"*, IEEE Trans. on Communications, Vol. COM-24, pp.1130-1139, October 1976 and pp.1238-1245, November 1976.

37. S A Fredricsson, *"Optimum transmitting filter in digital PAM systems with a Viterbi detector"*, IEEE Trans. on Information Theory, Vol . IT-20, pp.479-489, July 1974.

38. S A Fredricsson, *"Joint optimization of transmitter and receiver filters in digital PAM systems with a Viterbi detector"*, IEEE Trans. on Information Theory, Vol. IT-22, pp.200-210, March 1976.

39. A Contoni and K Kwong, *"Further results on the Viterbi Algorithm equalizer"*, IEEE Trans. on Information Theory, pp.764-767, November 1974.

40. F L Vermeulen, *"Low complexity decoders for channels with intersymbol interference"*, PhD Thesis, Stanford University, 1975.

41. G J Foschini, *"A reduced state variant of maximum likelihood sequence detection attaining optimum performance for high signal-to-noise ratios"*, IEEE Trans. on Information Theory, Vol. IT-23, No. 5, pp.605-609, September 1977.

42. J Gordon and N Montague, *"Channel equalization using a stack algorithm"*, IERE Conference Proceedings No. 37, pp.107-114, September 1977.

43. S A Fredricsson, *"A reduced state Viterbi detector for multilevel partial response channels"*, Technical Report No. 86, Telecommunication Theory, Royal Institute of Technology, Stockholm, Sweden, September 1974.

44. W U Lee and F S Hill, Jr., *"A maximum likelihood sequence estimator with decision-feedback equalization"*, IEEE Trans. on Communications, Vol. COM-24, No. 9, pp.971-979, September 1977.

45. J K Omura, *"On optimum receivers for channels with intersymbol interference"*, (abstract), presented at the IEEE International Symposium on Information Theory, Noordwijk, Holland, June 1970.

46. H Kobayashi, *"Correlative level coding and maximum likelihood decoding"*, IEEE Trans. on Information Theory, Vol. IT-17, pp.586-594, September 1971.

47. A P Clark, J D Harvey and J P Driscoll, *"Improved detection processes for distorted digital signals"*, IERE Conference Proceedings No. 37, pp.125-136, 1977.

48. A P Clark, J D Harvey and J P Driscoll, *"Near-maximum-likelihood detection processes for distorted digital signals"*, The Radio and Electronic Engineer, Vol. 48, No. 6, pp.301-309, June 1978.

49. G W Irwin and J P Driscoll, *"The effect of merging on a Viterbi based detection process"*, Internal Report, Department of Engineering Mathematics, Loughborough University, February 1978.

50. P L Meyer, *"Introductory probability and statistical applications"*, Addison-Wesley, 1971.